# Random Access Channel in UMTS

By


## Yashesh Kamlesh Buch





**DEPARTMENT OF ELECTRICAL ENGINEERING**

## Indian Institute of Technology Kanpur


**MARCH, 2003**

# RANDOM ACCESS CHANNEL IN UMTS

*A Thesis Submitted*

in Partial Fulfillment of the Requirements

for the Degree of

**Master of Technology**

*by*

**Yashesh Kamlesh Buch**

*to the*

**DEPARTMENT OF ELECTRICAL ENGINEERING**

INDIAN INSTITUTE OF TECHNOLOGY,KANPUR

March 2003

# CERTIFICATE

It is certified that the work contained in the thesis entitled "*Random Access Channel in UMTS*" by *Yashesh Kamlesh Buch* has been carried out under my supervision along with joint supervision of Prof. Dr. Bernhard H. Walke of the University of Technology, Aachen, Germany and that this work has not been submitted elsewhere for a degree.

March 2003

(V. Sinha)
Professor,
Department of Electrical Engineering,
Indian Institute of Technology,
Kanpur-208016.

## ACKNOWLEDGMENT

I am grateful to Dr. V. Sinha who provided me with an opportunity to persue my thesis at University of Technology, Aachen, Germany. His guidance and a continues urge to the students to do something innovative will go a long way in shaping my career.

I also express my grattitude to all the professors of IIT Kanpur, in particular, Dr. A. K. Chaturvedi and Dr. S. K. Bose for providing quality education and a great insight in diverse areas of communications.

I am obliged to the Chair of Communication Networks, Dr. Bernhard H. Walke at University of Technology, Aachen, Germany for giving me an opportunity to work for one of his projects. His periodic inputs and the help extended by research scholar Matthias Malkowski are invaluable.

Finally, I thank all my friends for giving the much needed moral support in some tough times during the course of the thesis.

YkBuch

Yashesh Kamlesh Buch

## ABSTRACT

The *Universal Mobile Telecommunications System (UMTS)*, which uses *Wideband Code Division Multiple Access* (WCDMA) radio interface, promises exceptionally high data rates as well as *Quality of Service* (QoS) features. The most important issue in fulfilling these promises is to use the available radio resource as efficiently as possible. It is envisaged that most typical multimedia applications will require the *User Equipment* (UE) to transmit only small amounts of data to make requests for starting sessions. Once the session is established most traffic flow would be downlink. Using dedicated channels for such applications is wasteful of resources. The *Random Access Channel* (RACH) is a common uplink channel used in the *Global System for Mobile Communication* (GSM) only for the purpose of call establishment and maintenance; however, in UMTS the RACH is conceived to be used more effectively for transmission of small amounts of dedicated data in asymmetric traffic scenarios. Hence throughput and delay characteristics of the RACH become a matter of close investigation.

The UMTS radio interface, in particular the *Random Access Channel* (RACH) is studied. The *Physical Random Access Channel* (PRACH) and the *Acquisition Indicator Channel* (AICH) have been modeled as specified in 3GPP standards. One of the main objectives is to enhance *UMTS Radio Interface Simulator* (URIS) which is under development at the Chair of Communication Networks (ComNets), RWTH, Aachen. Simulations have been performed with the modeled PRACH by varying load and other critical parameters. Throughput and delay characteristics of the RACH have been studied from these simulations. Throughtput of the RACH is found to be closely matching with that of multichannel slotted ALOHA.

# CONTENTS

## LIST OF FIGURES

# CHAPTER 1

# Introduction

For more than a decade research has been going on to find enabling techniques to introduce multimedia and high bit-rate data services to mobile communications. At the same time the demand for such applications, as well as the number of subscribers to cellular networks, has been growing permanently. As a result, *3rd Generation* (3G) mobile communication systems with high-performance radio access techniques are under development and being introduced in the market

In the near future *Universal Mobile Telecommunications System* (UMTS) users will be provided with data rates of up to 144 kbps in macro-cellular environments, up to 384 kbps in micro-cellular environments and up to 2 Mbps in indoor or pico-cellular environments. To meet these requirements an efficient use of the available spectrum by high performance radio access techniques is needed.

In UMTS, communication resources such as frequency, time and channelisation codes are shared amongst a large, unknown number of mobile terminals referred to as *User Equipment* (UE). Moreover, once a UE gets a resource somehow, it should possess it for the length of time it wants to communicate. Thus there are two phases of communication:

1. **Contention phase:**

   UEs compete for the resources on a common uplink channel called the *Random Access Channel* (RACH), using well known slotted ALOHA random access technique.

2. **Transmission phase:**

   At the end of the contention phase UEs manage to get a resource dedicated

to itself. It is relinquished only after a UE has completed communication.

The contention phase and the behaviour of corresponding RACH is covered in this thesis.

Throughput and delay characteristics of different random access schemes have always been a classical topic of research; particularly so, in case of UMTS because of two main reasons:

1. **The WCDMA air interface of UMTS:**
   This implies interference limited radio access networks and corresponding problems of power control.
2. **The use of RACH to carry dedicated data:**
   This is conceived for asymmetric, multimedia applications of UMTS For example, transmission of a *Uniform Resource Locator* (URL) to start a web browsing session.

The Chair of Communication Networks (ComNets) at the Aachen University of Technology is currently developing a simulator called the *UMTS Radio Interface Simulator* (URIS) for the radio interface of UMTS. It aims at performance evaluation of various parts of the UMTS protocol stack. An opportunity was provided by the chair to study, use and modify the simulator.

Within the framework of this thesis, various parts of the URIS related to the RACH have been improved. Most importantly, the *Physical Random Access Channel* (PRACH) and corresponding *Acquisition Indicator Channel* (AICH) which is used for sending acknowledgments have been modeled as specified in 3GPP standards. The random access situation is simulated by varying number of mobiles, traffic and RACH parameters. Implementation done in the thesis is validated. Throughput and delay characteristics of RACH in various interesting cases have been noted and discussed.

This introduction is followed by a brief overview of the UMTS radio interface protocol architecture in chapter 2. Chapter 3 gives a detailed account of RACH structures and procedures in MAC and Physical layers. The simulator URIS is described in chapter 4. Chapter 5 explains the implementation done in the thesis in detail. In Chapter 6, the RACH is compared with the classical slotted ALOHA and the results obtained from simulation in the thesis are discussed with reference to the same. A summary of the thesis and an outlook for further development is presented in chapter 7.

CHAPTER 2

# UMTS Radio Interface Architecture

This chapter briefly describes the architecture of the radio interface within the UMTS protocol. More details can be found in the 3GPP Technical Specification [13], [14] and [20]. Apart from the standards, a good explanation of this is provided in [1] and [4].

As shown in figure 2.1 within the UMTS architecture two main parts can be distinguished:

1. An *Access Stratum* (AS), consisting of the parts in the infrastructure and in the *User Equipment* (UE) and the protocols between these parts which are specific to the access technique.
2. A *Non-Access Stratum* (NAS) which is independent from the access technology.

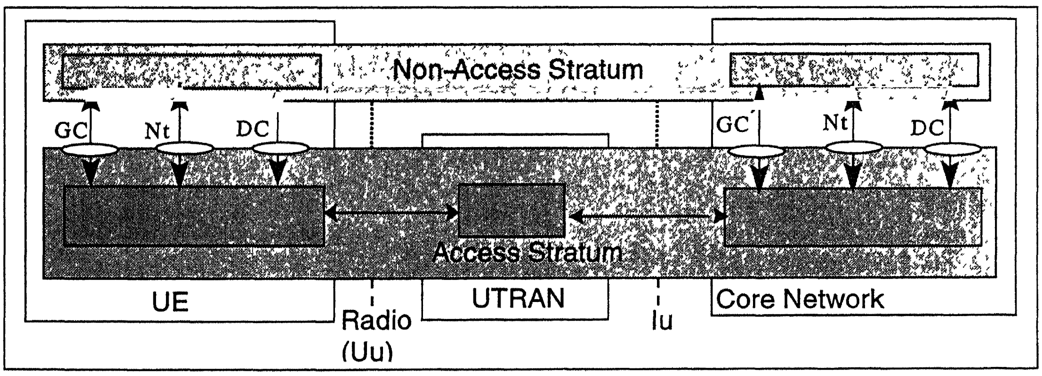



**Figure 2.1:** UMTS Architecture

The communication between these two parts relies on the *Radio Bearers* (RBs) offered by the Access Stratum using three different *Service Access Points* (SAPs):

the *Notification SAP* (Nt-SAP), the *General Control SAP* (GC-SAP) and the *Dedicated Control SAP* (DC-SAP). The Nt-SAP's purpose is to broadcast data to identified users, the GC-SAP is used to enable the Core Network to provide information and to send commands which are not related to a specific UE, the DC-SAP provides connection establishment and release to a specific UE.

On a functional basis the *Access Stratum* is divided in three layers, which are defined by the ISO/OSI (*International Organization for Standardization/Open Systems Interconnection*) reference model:

- Layer 3: *Network Layer* (NL)
- Layer 2: *Data Link Layer* (DLL)
- Layer 1: *Physical Layer* (PL or PHY)

A further classification can be done as follows:
A layer is said to be part of the *user plane* when it provides services for the transport of user data, or of the *control plane* when it deals with signaling for connection setup and release, and for radio bearer management.
Figure 2.2 depicts the layered structure within the *Access Stratum*.

The only protocol entity contained within the network layer is the *Radio Resource Control* (RRC), which belongs to the control plane. The major part of the control signaling between UE and *UMTS Terrestrial Radio Access Network* (UTRAN) are RRC messages, carrying all the parameters required to set up, modify and release layer 2 connections. Furthermore, the RRC has a direct connection to every entity, purpose of which is to exchange control information(i.e. it is used by the RRC to ask for measurements and by the lower layers to report measurements results or errors).

Below the network layer, the *Data Link Layer* contains four sub-layers:

- the *Packet Data Convergence Protocol* (PDCP),

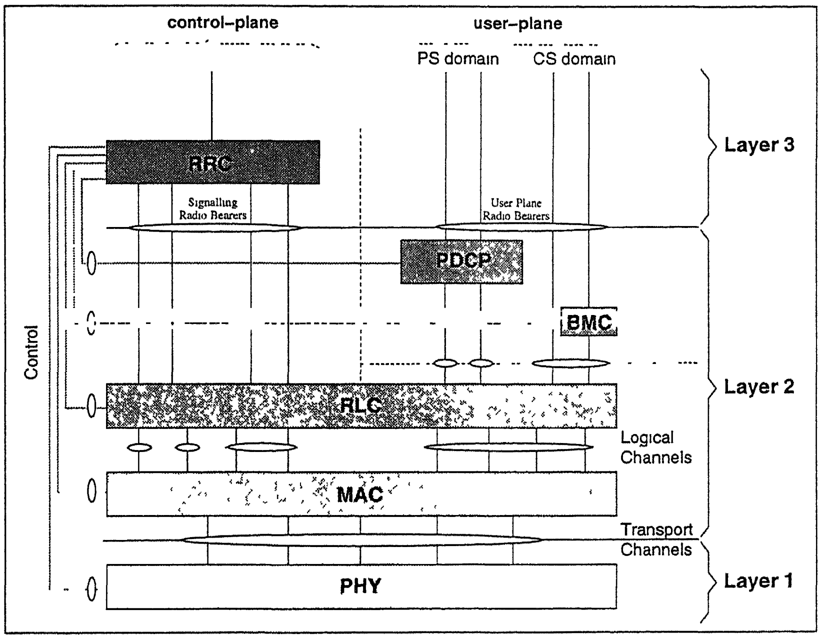



**Figure 2.2:** Radio Interface Protocol Stack

- the *Broadcast/Multicast Control* (BMC),
- the *Radio Link Control* (RLC) and
- the *Medium Access Control* (MAC).

The PDCP and BMC protocols exist in the user plane only. The former is used for services belonging to the packet switched domain, being responsible for the adaptation of radio network layer PDUs to the UMTS radio interface. Another PDCP's task is to increase channel efficiency by protocol header compression, for example.

The BMC is used for broadcast and multicast support, like *Short Message Services* (SMSs) and *Cell Broadcasts* (CBs). Both, PDCP and BMC offer *Radio Bearers* to higher layers.

The *Radio Link Control* protocol performs segmentation and retransmission services for both, user data and control information. It can be configured by the RRC to operate in one of three modes depending on the required *Quality of Service* (QoS): *Transparent Mode* (TR), *Acknowledged Mode* (AM) or *Unacknowl-*

*edged Mode* (UM).
The RLC belongs to both, control and user planes, so that its offered services are both, *Signalling Radio Bearer* (SRB) and *Radio Bearers* (RBs).

The *Medium Access Control* protocol performs the mapping of the *Logical Channels* (LoCHs) coming from RLC, onto the *Transport Channels* (TrCHs), which are offered to the physical layer. A set of logical channels is defined to transmit a specific type of information; therefore, each of them determines the kind of information it uses. On the other side, the transport channels describe how data is to be transmitted over the air interface and with what characteristics. These are specified by means of the *Transport Format* (TF).
The MAC is responsible for selecting an appropriate transport format for each transport channel depending on the instantaneous source rates on the logical ones. The selection is performed with respect to the currently active *Transport Format Combination Set* (TFCS), which is defined for each connection by the RRC admission control.

The *Random Access Channel* in MAC and physical layers is the topic of this master's thesis and is described in detail in the next chapter.

CHAPTER 3

# Random Access Channel in UMTS

This chapter describes the *Random Access Channel* (RACH) and the *Acquisition Indicator Channel* (AICH) in detail, including various structures and procedures that exist for these channels in physical and MAC layers. The specification, from which particular information about RACH is extracted, is referenced at the appropriate paragraphs in the chapter.

## 3.1 Evolution of RACH procedures

This section describes the evolution of random access procedure that is accepted as a standard today from the one in second generation mobile communication standards like GSM and IS-95.

In simple terms the RACH in UMTS is described as *a slotted ALOHA channel with fast acquisition indication.* Slotted ALOHA is a well known protocol for random access while meaning of the phrase *fast acquisition indication* will become clear in a detailed explanation of random access procedure to follow.

Traditionally, RACH could have been simply a slotted ALOHA channel like the one in GSM with open loop power control. A RACH packet is transmitted with some small initial power based on an estimate of path loss from a downlink channel. If no acknowledgment is received until a predetermined timeout, the packet is retransmitted with a power based on the current estimate of downlink path loss. This is repeated until a positive acknowledgment is received.

The problem with this simple scheme is that open loop power control is highly unreliable and a fading dip in the uplink can cause the RACH packet to be undetectable in presence of more powerful and close loop power controlled channels like *Dedicated Channel* (DCH).

A possible solution to this could have been transmitting the RACH packet with sufficient power compared to the downlink path loss estimate. This could overcome the fading and make RACH packet detectable, but it will then cause excessive interference to other channels like DCH.

A scheme called *Message Power Ramping* is a solution to both the above problems.
In this scheme the first packet is transmitted based on open loop power control while for each retransmission the power is increased or ramped up in small, fixed steps.

But even message power ramping scheme has two drawbacks:

1. A RACH message being quite long (10 ms or 20 ms), each unsuccessful RACH transmission causes *Multiple Access Interference* (MAI) to other users. This leads to a decrease in overall capacity of the system.
2. Delay of a RACH message transmission is also higher owing to large message lengths and acknowledgments that are received from higher layers.

A solution to this is a scheme called *Preamble Power Ramping.* Eventually this scheme is currently adopted as a standard in 3GPP specifications for RACH transmissions. In this scheme two phases of RACH transmissions can be distinguished:

1. In the first phase a short burst from physical layer called *preamble*, spanning approximately 1.067 ms is transmitted with some initial value of power based on open loop power control. Acknowledgment of preamble is a phys-

ical layer response with no intervention from higher layers and hence a fast response. Therefore, the physical layer entity responsible for sending acknowledgments is named *Acquisition Indicator Channel* (AICH) in UMTS. For all subsequent retransmissions of the *preamble* the power is gradually increased or ramped up in fixed steps.

2. The second phase starts only after a positive acknowledgment of *preamble* on AICH. In this phase a 10 ms or 20 ms RACH message part is transmitted, a fixed time after the transmission of successful *preamble*.

With this scheme, the probability of the message getting through is largely improved. The preamble power ramping effectively provides some sort of carrier sensing mechanism prior to actual message transmission improving the overall throughput of RACH transmissions compared to the traditional slotted ALOHA. Moreover, as the preamble is much smaller in length than the actual RACH message, interference and delay are also notably reduced.

## 3.2 Information flow from higher layers

The flow of information from higher layers up to the physical layer for a RACH is as shown in figure 3.1.

Following three types of logical data can be mapped to the transport channel RACH.

1. Control data used for call establishment from logical *Common Control Channel* (CCCH). ( The usual function of a RACH )
2. Specific user's traffic data from logical *Dedicated Traffic Channel* (DTCH).
3. Specific user's control data from logical *Dedicated Control Channel* (DCCH). ( 2 and 3 can be considered to be additional functions of RACH )

The transport block or MAC PDU of RACH is further mapped to its physical counterpart PRACH's data part. MAC-C entity takes care of this in MAC

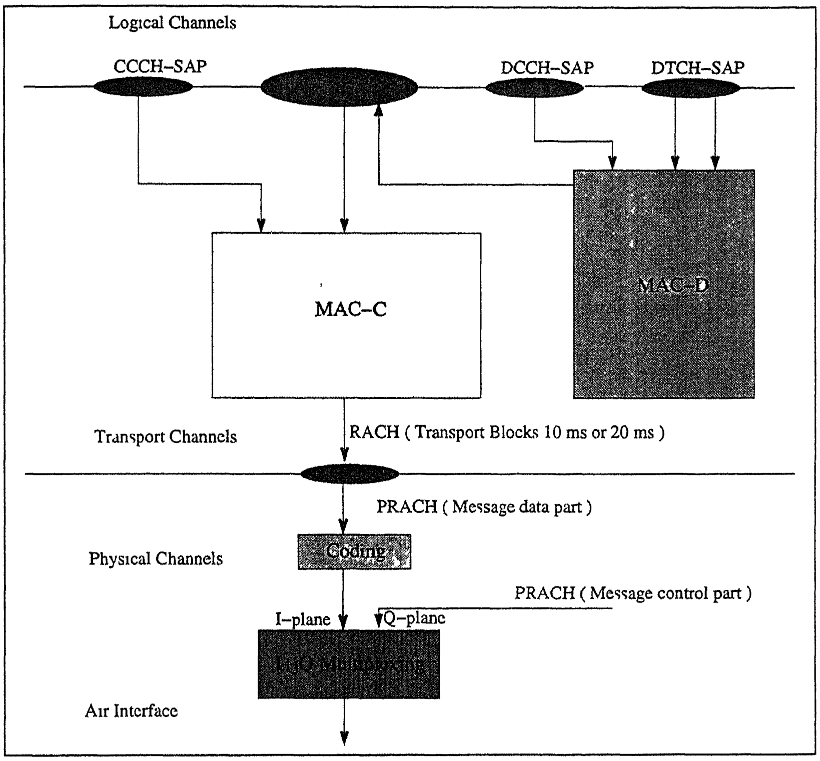



**Figure 3.1:** Flow of RACH information from higher layers

Further ahead in physical layer, the data part is first coded and then spread on I-plane. Thereafter, it is combined with the physical layer control information like pilot and *Transport Format Combination Indicator* (TFCI) bits which have been spread in Q-plane. Finally,the combined information is transmitted over the air interface.

It should be noted that any type of logical data in RACH transport blocks will be transmitted only after the successful completion of preamble phase as discussed in section 3.1.

## 3.3 Physical layer structures

Information in this section is compiled from [16]

### 3.3.1 Slot structure for slotted ALOHA algorithm

Since RACH employs slotted ALOHA, both, the preambles and the messages are transmitted at the beginning of well-defined time intervals called *Access Slots (AS)*. For this purpose, it is assumed that the UEs are in sync with the *UMTS Terrestrial Radio Access Network* (UTRAN).

As shown in figure 3.2, a 20 ms period (equivalent to two radio frames) is divided into 15 access slots. Thus duration of each slot is approximately 1.33 ms or 5120 chips. Timing relationship of these access slots in PRACH with respect to AICH will be discussed in section 3.3.5.

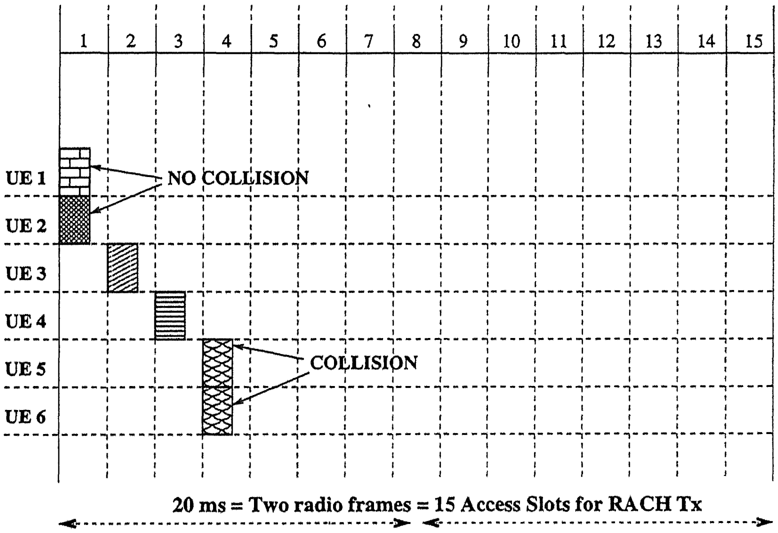



**Figure 3.2:** Slot structure for slotted ALOHA transmissions

The random access transmission consists of one or several preambles of length 4096 chips followed by a message of length 10 ms or 20 ms.

### 3.3.2 RACH preamble part

- Length 4096 chips
- Each preamble is a complex valued sequence built from a scrambling code and a signature.
- A preamble scrambling code is constructed from long scrambling sequence and bears a one to one correspondence with the scrambling code used in downlink channels of the cell.
- A preamble signature is made up of 256 repetitions of a length 16 Hadamard code.

  ( 256 x16 = 4096 )

  Since there are 16 Hadamard codes of length 16, 16 signatures are available for forming a RACH preamble.
- It can be summarised that different preambles in a UMTS cell are constructed from same scrambling code, but one, out of 16 different signatures. Thus preambles can be differentiated simply according to the signature.
- Specific equations for complex valued sequence, scrambling code and signatures are not very important from the view of this thesis and hence have been omitted for the sake of brevity. Interested readers can refer to [14].
- Preamble part is purely a physical layer burst structure.
- Preamble acknowledgment is also limited to physical layer signaling and is thus fast.

### 3.3.3 RACH message part

- Length: One or two radio frames, i.e.10 ms or 20 ms.
- The 10 ms message part radio frame is split into 15 slots. each slot of length $T_{slot} = 2560$ chips
- Like any other physical channel in UMTS, each slot of PRACH consists of two parts, a data part and a control part, which are transmitted simultaneously in time. but orthogonally on I and Q planes respectively.

- The data part consists of the transport block set mapped from the transport channel RACH.
- The control part carries layer 1 control information like pilot bits and the TFCI for the corresponding data part.
- A detailed diagrammatic representation of data and control parts of PRACH per slot is shown in figure 3.3.

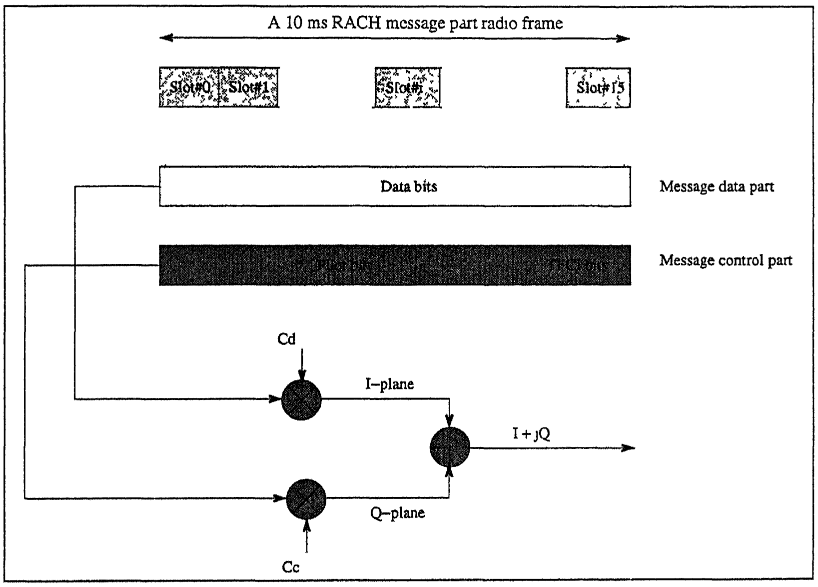



**Figure 3.3:** RACH message part

- As mentioned in section 3.3.2 details of channelisation and scrambling codes for RACH message part are not of relevance to the current work and hence omitted. However, the following about channelisation codes is important to note.
  Suppose, a preamble signature is formed from S, a Hadamard sequence of length 16. This S points to one of the 16 nodes in *Othogonal Variable Spreading Factor* (OVSF) code-tree. The subtree below the node corresponding to S is used for spreading the RACH message part. Thus the *Spreading Factor* (SF) for the RACH data part can only be 32, 64, 128 or 256.
- Message data part is nothing but *Transport Block Set* (TBS) from MAC (MAC PDU) which in turn has a mapping from a logical channel of RLC.

- Hence, the acknowledgment of the message part involves higher layer signaling and is thus slower.

### 3.3.4 AICH burst

- A fixed rate (SF = 256) physical channel used to carry *Acquisition Indicators ( AI )*.
- As shown in figure 3.4, AICH consists of a repeated sequence of 15 consecutive access slots, each of length 5120 chips.
- Each access slot consists of two parts, an *Acquisition Indicator ( AI )* part consisting of 32 real-valued symbols $a_0$ to $a_31$ and a part of duration 1024 chips without any transmission. This part is not formally a part of the AICH but reserved for possible use by *CPCH Status Indicator Channel* (CSICH) or possible future use by other physical channels.

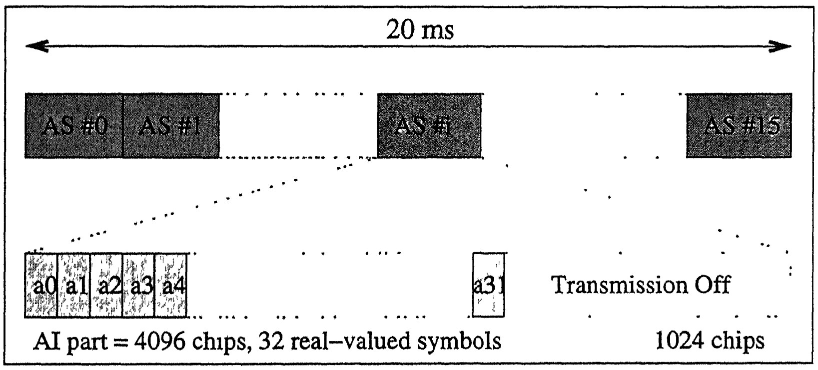



**Figure 3.4:** Structure of *Acquisition Indicator Channel* (AICH)

- The real valued symbols $a_o$, $a_1 .... a_{31}$ in the figure are given by

$$a_j = \sum_{s=0}^{15} AI_s b_{s,j}$$

where $AI_s$ is an acquisition indicator corresponding to signature S and $b_{s,0}$ to $b_{s,31}$ is defined in the specifications.

- If an AI is set to +1, it represents a *Acknowledgement* (ACK) while if it is set to -1, it represents a *Negative Acknowledgement* (NACK).

- The real valued symbols, $a_j$, are spread and modulated in the same fashion as bits when represented in $\{+1, -1\}$ form.

### 3.3.5 PRACH-AICH timing relationship

This is shown in the figure 3.5 below.

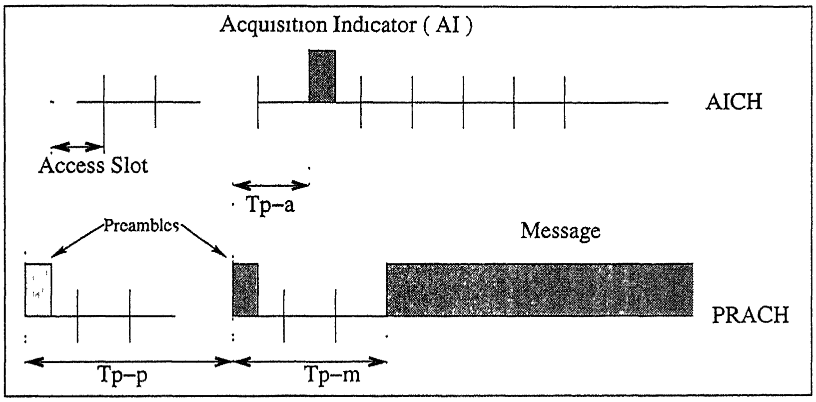



**Figure 3.5:** Timing relationship between PRACH and AICH as seen at the UE

If a parameter AICH-Transmission-Timing signaled by higher layers is set to 0:

$T_{pp,min}$ = 15360 chips = 3 access slots ( minimum time )

$T_{pa}$ = 7680 chips = 1.5 access slots ( fixed time)

$T_{pm}$ = 15360 chips = 3 access slots ( fixed time )

If AICH-Transmission-Timing is set to 1:

$T_{pp,min}$ = 20480 chips = 4 access slots ( minimum time )

$T_{pa}$ = 12800 chips = 2.5 access slots ( fixed time )

$T_{pm}$ = 20480 chips = 4 access slots ( fixed time )

The above section completes the discussion of access slot structure in physical layer for RACH transmissions, structure of various physical layer bursts viz RACH preamble, RACH message and AI and the timing relationship between PRACH and AICH.

The next sections explain RACH procedures in MAC and physical layers. Since the RACH procedure starts in MAC and then carries on in physical layer, the former is discussed first followed by the later. But as such the procedures are inter-related and not independent of each other.

The following figure 3.6 describes the complete RACH procedure encompassing both the MAC and physical layers and the primitives exchanged between them It gives a good overview of the combined procedure.

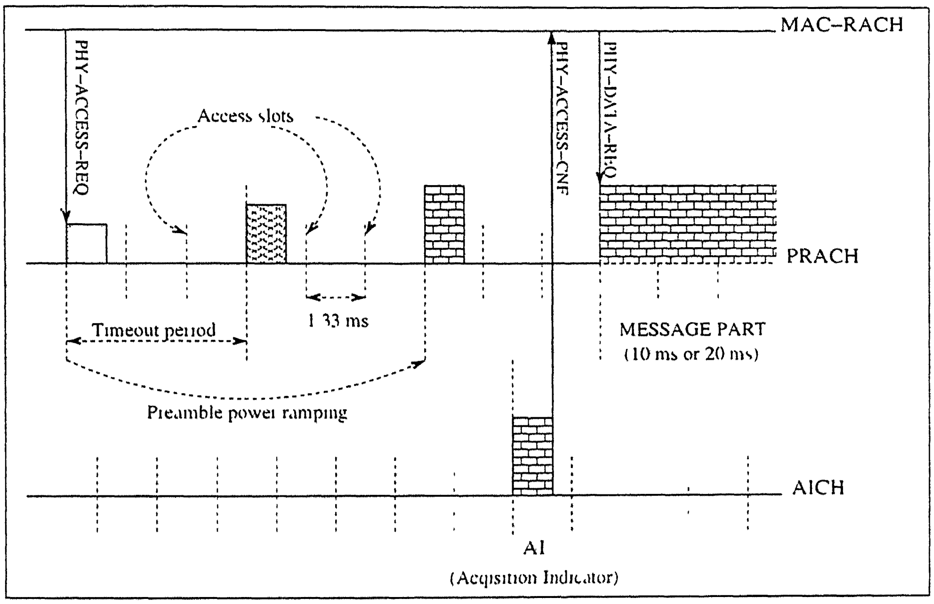



**Figure 3.6:** Combined MAC-PHY RACH procedure in nutshell

MAC sub-layer is in charge of controlling the timing of RACH transmissions on TTI level, while the timing on access slot level is controlled by the physical layer

## 3.4 RACH procedure in MAC

Information in this section is compiled from [17]

### 3.4.1 Access Service Class (ASC) selection

The concept of access service class is to support the QoS features in UMTS Different ASCs are assigned different resources or different priorities of usage of common resource When a particular logical or transport channel is said to be in an access service class, it has a certain priority with respect to other logical or transport channels while using the underlying physical layer resources.

Specifically, in case of RACH, physical layer resources, namely access slots and preamble signatures are divided between different ASCs. But at the same time it is also possible for more than one ASC or for all ASCs to be assigned to the same $\{accessslot, signature\}$ space. In such cases the ASC with higher priority is assigned a higher persistence value for the slotted ALOHA random access algorithm.

According to the specifications, an access service class is specified by a pair $\{i, p_i\}$. where i defines a certain partition of PRACH resources and $p_i$ defines the persistence value. There can be a maximum of 8 ASCs. The Lower the value of i, the higher is the priority. Thus ASC 0 results in the highest priority.

The selection of an ASC is a higher layer procedure and not within the scope of this thesis. The RACH procedure in MAC requires just an array with persistence values, indexed by ASC. This array is forwarded from RRC to MAC in the CMAC-Config-Req primitive.

It may be noted that the discussion of ASC here is only for the sake of completeness. The current version of simulator has only one ASC with persistence value 1

### 3.4.2 Control of RACH transmissions in FDD mode

This procedure is very well explained in the specification [17] by a flow chart.

Following steps can be identified from the flow chart.

- MAC receives following three 'RACH Transmission Control Parameters' from RRC in the CMAC-Config-Req primitive.
  1. Array with indexes i and elements $p_i$.
  2. $M_{max}$, the maximum number of preamble ramping cycles.
  3. $Nbo_{1,max}$ and $Nbo_{1,min}$ , the maximum and minimum count of 10 ms intervals for back-off interval timer $Tbo_1$, which is set when a NACK is received on AICH.

  When any of these three is updated, RRC sends the CMAC-Config-Req primitive with a new set of 'RACH Transmission Control Parameters'.
- As soon as the MAC receives data to be transmitted from logical channels with the primitive MAC-Data-Req from higher layers, it selects the ASC from the available array $\{i, p_i\}$.
- Based on the selected ASC and its persistence value $p_i$, a UE decides whether to start the 'Physical PRACH Transmission Procedure' in the present TTI or not. For this purpose a persistency check is performed by selecting a random number between 0 and 1 and comparing it with the $p_i$ value. If the selected number is smaller than $p_i$ transmission is allowed otherwise new persistency check is performed in the next TTI. The persistency check goes on continuously until the selected random number is smaller than $p_i$ and transmission is allowed.
- Once a persistency check goes through, 'PRACH Transmission Procedure' is triggered by passing the primitive PHY-Access-Req to the physical layer. The 'PRACH transmission procedure' starts with a preamble power ramping cycle at the end of which a preamble acknowledgment information transmitted on the AICH is sent to the MAC RACH procedure with the primitive PHY-Access-Conf.
- There are three options for the preamble acknowledgment information and for each one different actions are taken in the MAC procedure:

1. **NoACK** = No Acknowledgment received

   The value of preamble ramping cycle count M is increased by one.

   If the incremented value is less than or equal to $M_{max}$ a new persistency check is performed for restarting the 'PRACH procedure' at the start of a new TTI. The alignment with the TTI is realised by a timer T2.

   If the incremented $M > M_{max}$, the 'MAC procedure' exits and a corresponding indication of RACH transmission failure is sent to higher layers.

2. **NACK** = Negative Acknowledgment

   In this case again M is incremented by one.

   If the incremented $M \leq M_{max}$ , the transmission is randomly backed off apart from performing a persistency check as in NoACK case.

   Again if $M > M_{max}$ the 'MAC procedure' exits and a failure indication is given to higher layers.

3. **ACK** = Positive Acknowledgment

   A request for RACH message data part transmission is sent to the physical layer with the primitive PHY-Data-Req which also carries the TBS from MAC sub-layer.

   Once the TBS from the MAC is received by physical layer it is up to physical layer to apply appropriate coding, interleaving and rate-matching and transmit the data along with control information over the air interface.

   Finally a successful completion of MAC transmission control procedure is indicated to higher layers.

Before each new persistency check 'RACH transmission control parameters' are updated, i.e.a check is done, whether CMAC-Config-Req with new parameters has come. Always the latest parameters are used in each TTI.

It should also be noted that erroneously received RACH message part is detected

by higher layers (RLC or RRC) and retransmission thereof is the responsibility of these layers

Following is the description of the 'PRACH transmission procedure' which is triggered during the MAC procedure by PHY-Access-Req primitive.

## 3.5 Physical random access procedure

Information in this section is compiled from [15]

As mentioned previously, this procedure is initiated by MAC with a PHY-Access-Req primitive after the initial persistency check with selected ASC parameters $\{i, p_i\}$ goes through.

Before the actual start of the procedure physical layer should have received following parameters from the RRC:

1. Preamble scrambling code, $Sr_{pre,n}$
2. Message length, 10 ms or 20 ms
3. AICH-Tx-Timing parameter, 0 or 1
4. Set of available signatures and available RACH sub-channels for each ASC
5. Power ramping step size for preamble, $\Delta\ P$
6. The maximum allowed number of preamble retransmissions, *Preamble Retransmission Max.*
7. Initial value of preamble power, *Preamble Initial Power.*
8. Power offset of the message part with respect to preamble,
   $P_{p-m} = P_{msg-contorl-part} - P_{preamble}\ in\ dB.$
9. Set of *Transport Format* (TF) parameters which includes power offset between message data part and message control part for each TF.

All the above mentioned parameters may be updated from higher layers before initiation of a new PRACH procedures.

At each cycle of PRACH procedure the MAC shall transfer the following parameters to the physical layer.

1. The Transport Format to be used for the PRACH message part.
2. The ASC of the PRACH transmission.
3. The data to be transmitted (*Transport Block Set* (TBS) ).

The physical random access procedure shall then be performed as follows:

1. Derive the available uplink Access Slot(AS) in the next full AS set, for the set of available RACH sub-channels within the given ASC. Randomly select one AS considering the AS as a uniformly distributed random variable.
   If there exists no AS in the current full AS set, try again in the next full AS set.
2. Randomly select a signature S from the set of available signatures within the given ASC, considering the signature as a uniformly distributed random variable.
3. Set the Preamble Retransmission Counter to *Preamble Retrans Max*
4. Set the parameter Commanded Preamble Power $P_{Commanded}$ to *Preamble Initial Power.*
5. Set Preamble Transmission Power $P_{Tx}$ as follows:
   a) $P_{Tx} = P_{max}, \, if \; P_{Commanded} > P_{max}$
   b) $P_{Tx} = P, \, where \, P \geq P_{Commanded} \; and \, P \leq P_{min}, \, if \; P_{Commanded} < P_{min}$
   c) $P_{Tx} = P_{Commanded}, \, if \; neither \; of \; a) \; and \; b) \; is \; true$
6. Transmit the preamble using the selected uplink access slot, signature and preamble transmission power.
7. If no positive or negative *Acquisition Indicator* (AI) (AI neither +1 nor -1) corresponding to the selected signature is detected following actions are taken:
   a) Select the next available AS in the set of available RACH sub-channels within the given ASC.

b) Randomly select a new signature from the set of available signatures within the given ASC as in the earlier case of signature selection.

c) Increase the Commanded Preamble Power by $\Delta P$, Power Ramp Step [dB]. If the Commanded Preamble Power exceeds the maximum allowed power by 6 dB, the UE may pass L1 status ('NoACK on AICH') to the MAC and exit the physical random access procedure.

d) Decrease the Preamble Retransmission Counter by one.

e) If the Preamble Retransmission Counter $> 0$ then repeat from step 5. Otherwise pass L1 status ('NoACK on AICH') to the MAC and exit the physical random access procedure.

8. If a negative acquisition indicator corresponding to the selected signature is detected in the downlink access slot, pass L1 status ('NACK on AICH received') to the MAC and exit the physical random access procedure.
9. Finally, if a positive acquisition indicator is received, transmit the random access message three or four uplink access slots after the uplink access slot of the last transmitted preamble depending on the *AICH transmission timing* parameter. Transmission power of the control part of the random access message should be $P_{p-m}$ dB higher than the power of the last transmitted preamble. Transmission power of the data part of the random access message is set according to the specifications.
10. To end the procedure pass L1 status 'RACH message transmitted' to the higher layers and exit the physical random access procedure.

The physical random access procedure as specified in 3GPP and repeated above for the sake of continuity is far more elaborate than the one implemented in this thesis. Intricacies like ASC selection, finding out the set of available signatures and access slots as well as the concept of RACH sub-channels have not been modeled in the thesis.

A flow chart of a more simpler version of the physical random access procedure

which is implemented in the thesis is given in figure 3.7 below.

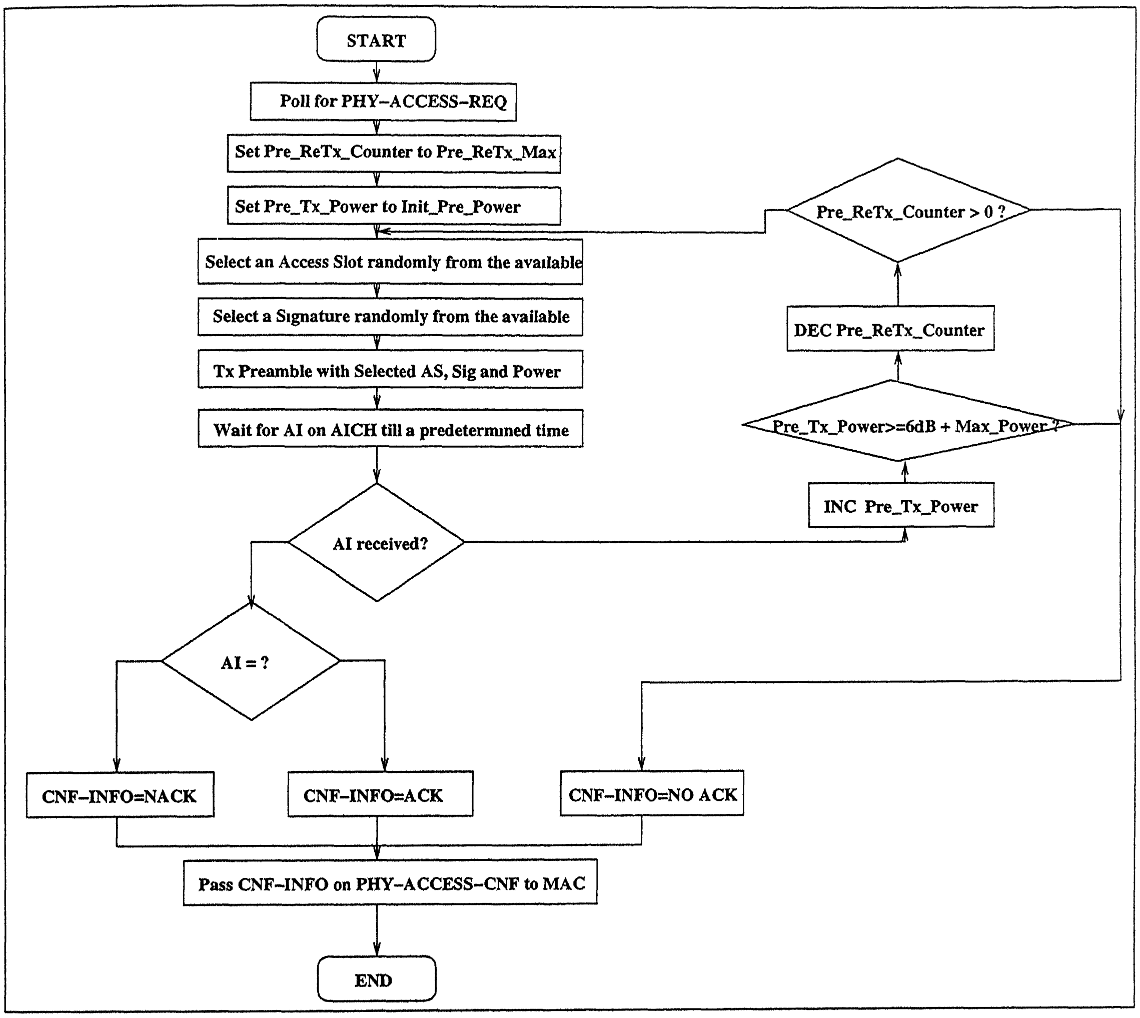



**Figure 3.7:** Simplified physical random access procedure

CHAPTER 4

# UMTS Radio Interface Simulator

This chapter describes the structure of the *UMTS Radio Interface Simulator* (URIS). The URIS is an event driven simulation environment which is composed of different functional modules, partly implemented in SDL and C++. The SDL part contains the UMTS protocol stack, while C++ provides classes for the simulator environment and also for modeling the channel, the positioning and the radio propagation. A detailed description on the URIS can be found in [5].

The C++ code is based on the *SDL Performance Evaluation Tool Class Library* (SPEETCL). The *SDL2SPEETCL* code generator provides the automatic conversion between parts programmed in SDL to C++/SPEETCL code. Both the class library and the SDL to SPEETCL converter have been developed at the Chair of Communication Networks (see [7] and [8]).

## 4.1 Simulator Structure

The general structure of the URIS is shown in figure 4.1. Every *User Equipment* (UE) and *Radio Network Controller* (RNC) is represented by one instance of an SDL system. All SDL system instances are enclosed by the same SDL environment. This environment represents the interface to other simulator modules written in C++.

The *load generators* are one of those parts. They are placed on top of the protocol stacks of the communicating entities and create the traffic needed for the simulation. Up to now, the communication relationship is always between one

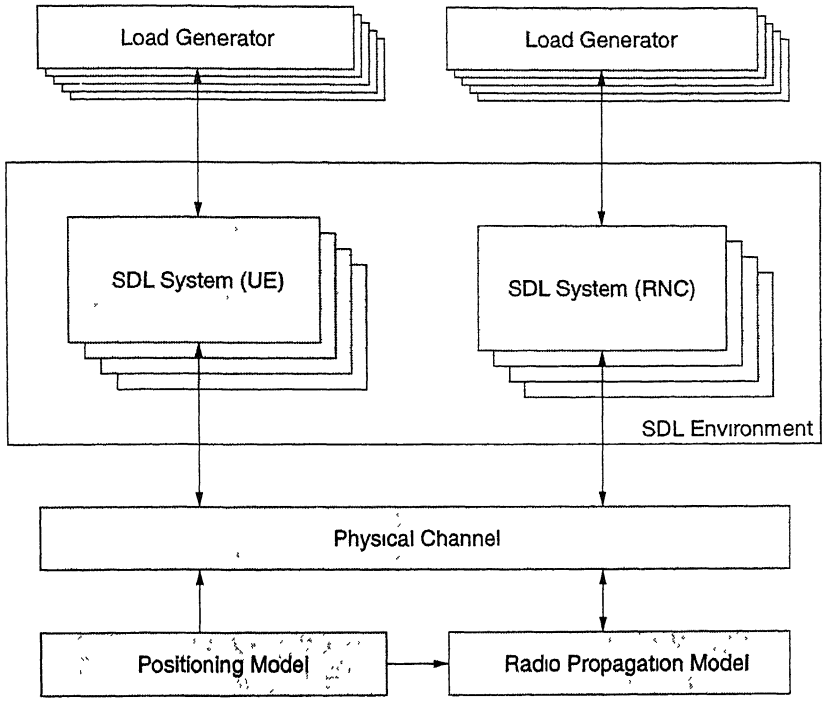



**Figure 4.1:** URIS Simulator Structure

*User Equipment* and one *Radio Network Controller*.
Placed below the SDL systems, the *physical channel* is a module in charge of the simulation of the actual radio transmissions. For example, the loss of transmitted data can be simulated. The simulation of the physical channel can be improved by a *positioning model* and a *radio propagation model*. Both are not integrated into the simulator yet.

The SDL systems use, like any other event driven module in the simulator, a global scheduler to send SDL signals and to set timers. With the definition of new SDL types, it is possible to access and use C++ and SPEETCL classes as external *Abstract Data Types* (ADTs). In this way all benefits of the SPEETCL like container classes and PDU assembly are available in SDL.
Within each SDL system the UMTS (sub)layers *physical layer*, *Medium Access Control*, *Radio Link Control*, *Packet Data Convergence Protocol* and *Radio*

*Resource Control* can be found as SDL blocks. The *Broadcast/Multicast Control* (BMC) Protocol is not implemented within the URIS so far. Figure 4.2 shows the structure of the SDL system.

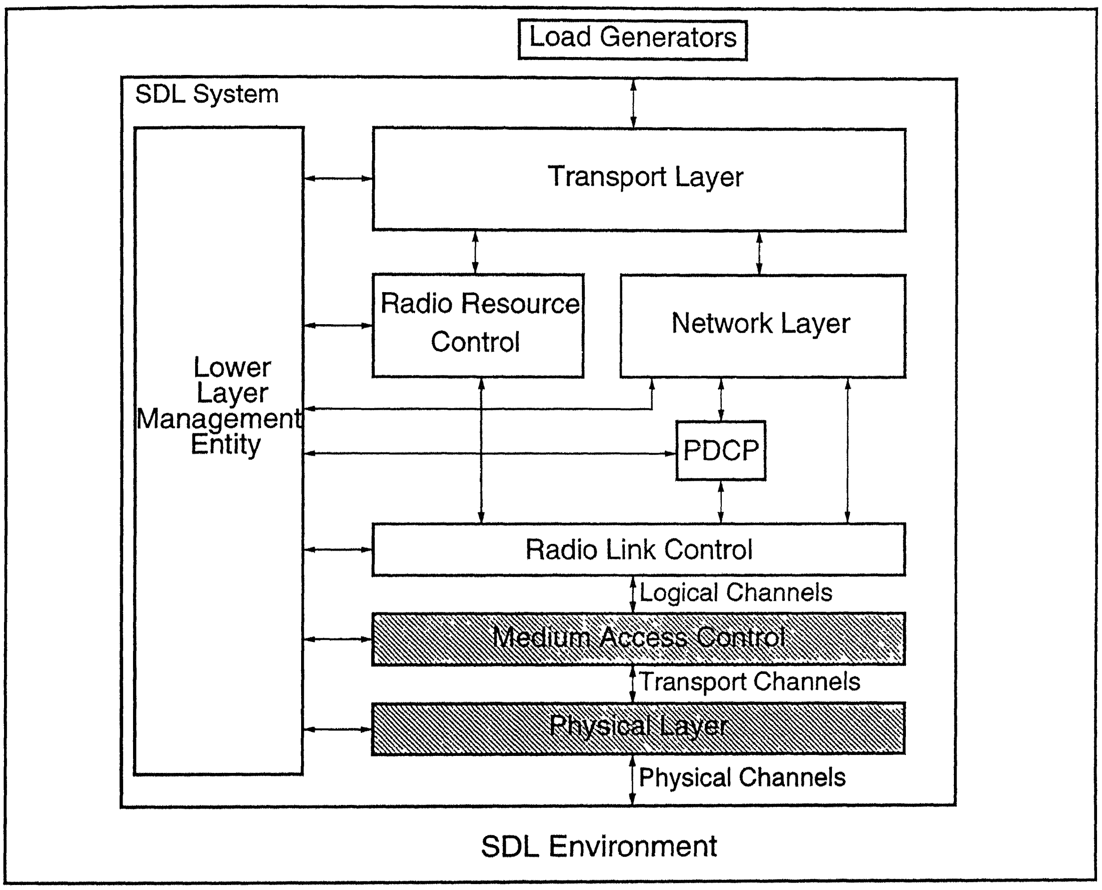



**Figure 4.2:** General UMTS Protocol Structure in SDL

The *Network Layer* (NL) provides *Internet Protocol* (IP) services to the *Transport Layer* (TL), which contains the transport protocols *Transmission Control Protocol* (TCP) and *User Datagram Protocol* (UDP). Both network and transport layers are not part of the UMTS radio interface protocols, but they have been implemented into the simulator to analyze the interaction between the protocols of the different layers and to evaluate the overall performance of the UMTS System experienced by the user.

In addition to the *Service Access Point* (SAP) for peer-to-peer services the UMTS

protocol specifies an inter-layer communication from the RRC protocol to the other UMTS protocol layers for configuration and measurement purposes. In the URIS this communication is executed through the *Lower Layer Management Entity* (LLME) block ([5]). The connection between LLME and the different protocol layers corresponds to the control SAPs in the UMTS specification (see chapter 2).

### 4.1.1 Inter-Layer Communication

A protocol process extends received *Service Data Units* (SDUs) by protocol specific information. In general a protocol layer adds a header, i.e. a *Protocol Control Information* (PCI) field. The PCI and the SDU are then combined to a new *Protocol Data Unit* (PDU). Before the PDU is forwarded to the lower layer, it is assembled into an *Interface Data Unit* (IDU), consisting of an *Interface Control Information* (ICI) field and the PDU itself.

The ICI contains informations to coordinate the inter-working between both layers. The encapsulated data unit is sent to the peer entity. At the receiving side the PDU is disassembled in order to recover the SDU, which is stored inside. For a better understanding of the PDU assembling/segmentation mechanism, figure 4.3 shows the OSI inter-layer communication as proposed by the OSI reference model ([4]).

The SPEETCL class library provides definitions for PDUs and PCIs. The URIS extends these basic types with additional information. This is done by redefining the C++ classes. The new derived classes contain the desired information and methods to access the information. Additionally, it is necessary to introduce the new classes with their methods to SDL. This is done by defining *Abstract Data Types*, i.e. only the interface is provided that describes its operators and parameters. The implementation details need not to be known to the SDL system.

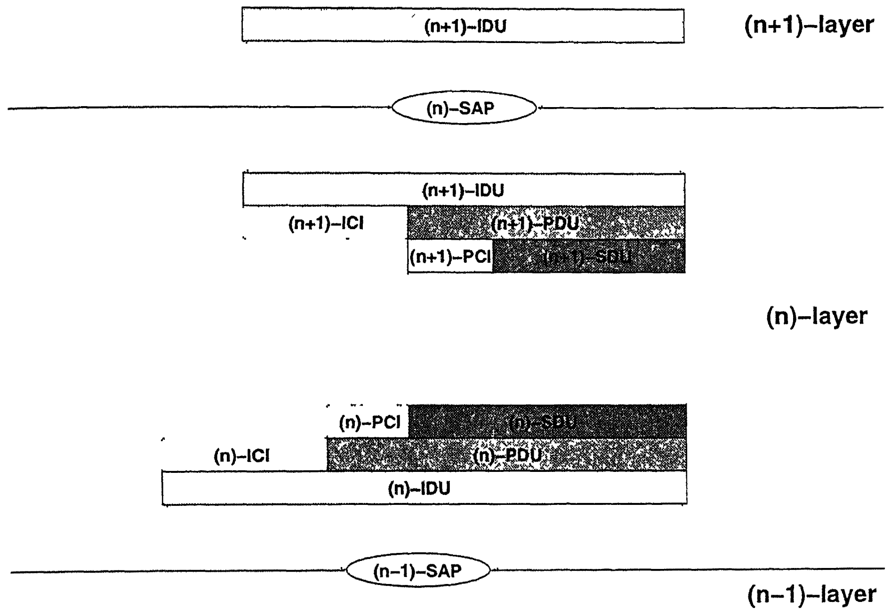



**Figure 4.3:** Inter-Layer Communication

The SPEETCL PDU contains only functionality for PCI and SDU combination and PDU identification. For the simulator additional informations are needed which are not part of the OSI definition of a PDU. These informations have been added to the PDU type already existing in the SPEETCL. The URIS extends the standard PDU type with additional information to the specialized *URIS PDU* (UPDU) in order to distinguish several connections:

- Manager Identifier (ManId),
- Connection Endpoint Identifier (CEPId),
- System Identifier (SysId).

The peer entity (receiver) can determine the sender of the PDU and can thus send a response. Additionally, the *URIS PDU* contains length information:

- PCI length,
- User Data (SDU) length,

The resulting PDU length is calculated from the PCI and SDU length. This information is necessary for segmentation and reassembling of PDUs. In addi-

tion to the inherited functionality, the *URIS PDU* supports methods needed for performance evaluation, e.g. the time when a UPDU is created and the type of service (speech, HTTP, etc.). Thereby, it is possible to calculate the transmission time of a PDU and thus the throughput of the protocols. Figure 4.4 shows the structure of the *URIS PDU*.

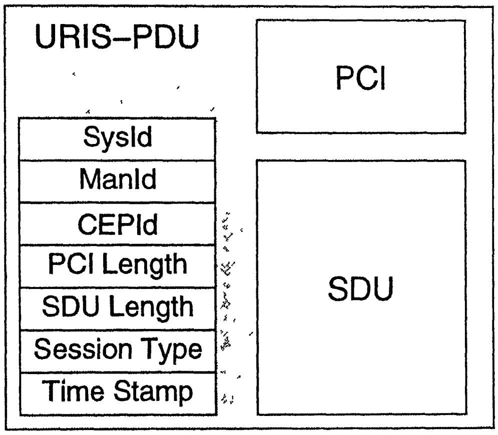



**Figure 4.4:** URIS Protocol Data Unit

The URIS assembles an *URIS PDU* into an *Interface Data Unit* (IDU) containing the *Interface Control Information* (ICI). The ICI contains only a *Connection Endpoint Identifier* (CEPId).

### 4.1.2 Object Inheritance

Each SDL system, representing the protocol stack of one RNC or UE described in section 4.1, is composed of a hierarchical tree of SDL system types. In figure 4.5 the inheritance hierarchy of these types is shown.

The root of the inheritance tree is the abstract type stOSI. It contains the blocks, processes, and procedures and also gates, signal definitions, and signal routes needed for a generic implementation of one protocol layer of the OSI reference model.

Derived from this base class the system type stUmts adds the functionality com-

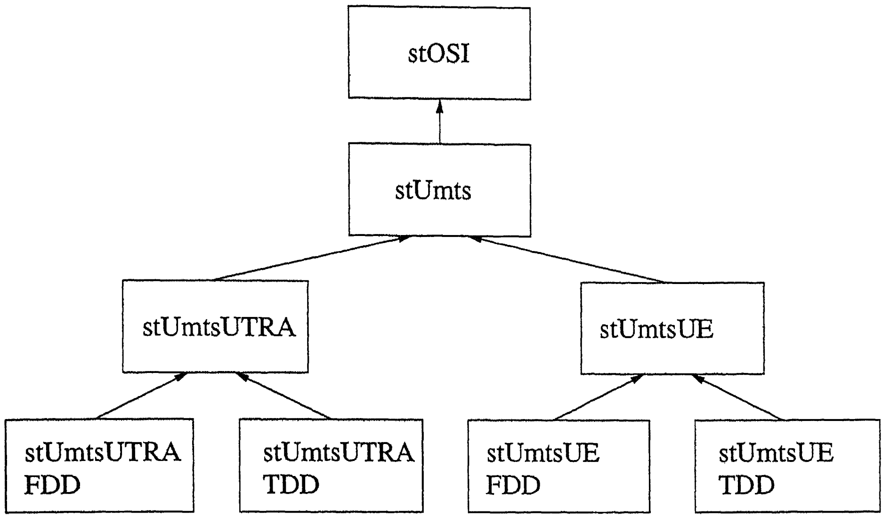



**Figure 4.5:** SDL Inheritance Tree

mon in all UMTS systems, i.e. UEs and RNCs. If functions differ between mobiles and RNCs these functions have to be implemented in the derived subclasses stUmtsUTRA (RNC) and stUmtsUE (UE).

Deviations in the implementation of the TDD and FDD mode have to be handled in the system types at the leaves of the inheritance tree.

## 4.2 Traffic Generators

In order to feed the simulator with communication load, traffic generators are implemented on top of the SDL systems. The traffic generators produce data traffic with the characteristics of typical applications.

For this purpose, the traffic generators from the SPEETCL class library are integrated into the URIS (see figure 4.1). The following traffic generators are supported and can be configured for specific simulation scenarios:

- Constant Bit Rate (CBR),
- Conversational Speech,
- Simple Mail Transfer Protocol (SMTP),
- Telnet,

- File Transfer Protocol (FTP),
- Hypertext Transport Protocol (HTTP).

Since the URIS provides simulation scenarios for circuit switched (CS) and packet switched (PS) networks, two independent traffic administrators are integrated. Every administrator manages a configurable pool of traffic generators. These load-mix tables provide high flexibility of the simulation scenarios.

## 4.3 Channel Model

The URIS has been developed to analyze the performance of the UMTS protocol stack. The physical layer applies error correction and detection for the transmitted data. Three different scenarios can be evaluated if bit errors appear during a transmission.

- The physical layer detects bit errors and is able to correct these errors. The layers above the physical layer don't experience any errors for their PDUs.
- The physical layer detects bit errors but is not able to correct these errors. This leads to missing *Transport Blocks* within a received *Transport Block Set*. A TB directly represents a RLC PDU.
- The physical layer is not able to detect an incorrect received burst. The received (corrupted) TBs are passed to higher layers where PDU failure could possibly be detected by checksum incoherence.

The physical layer is not fully implemented yet. A block error rate can be chosen and MAC transport blocks are dropped randomly with the configured error probability. Since every RLC PDU is mapped to a single MAC transport block, the configurable block error rate is corresponding to the rate of dropped RLC PDUs.

CHAPTER 5

# Implementation

This chapter provides a detailed description of the *Random Access Channel* (RACH) implementation within the physical layer, which has been developed in this thesis for the *UMTS Radio Interface Simulator* (URIS). Some tutorials on *Specification Description Language* (SDL), which has been used for a major part of implementation, can be found in [3] and [10]. [7],[8],[9] are also worth refering for any implementation dealing with the use of SDL and SPEETCL.

## 5.1 Status of RACH in URIS

The status of RACH implementations in URIS at the start of this thesis is briefly described below.

On the *User Equipment* (UE) side only the RACH transmission procedure in the MAC layer was implemented according to the specifications. This procedure passed the primitive PHY-Access-REQ to the physical layer RACH-handler to trigger the corresponding RACH procedure in the physical layer. But this RACH-handler lacked any such procedure. Hence, PHY-Access-REQ was responded simply by passing PHY-Access-CONF primitive with positive acknowledgment back to MAC layer. As a result, the MAC procedure continued and the PHY-Data-REQ was passed to the physical layer RACH-handler, which in turn lead to the transmission of the RACH message part.

The *UMTS Terrestrial Radio Access Network* (UTRAN) side was even simpler. RACH message part bursts were simply collected and sent to higher layers.

Thus there was no contention for the medium and hence no collisions. As such there was no random access mechanism taking place in the simulator. It can be said that RACH was existent only for the name sake with no real functionality at all.

Furthermore, as the simulator is currently functional only for *Frequency Division Duplex* (FDD) mode transmissions burst concept was only conceptual and *Transport Block Set* (TBS) from transport channels were transmitted in the form of whole radio frames aligned to TTI. Thus there was no provision for transmission of actual physical layer bursts like RACH preambles or *Acquisition Indicator* (AI)s which are not aligned to TTI.

## 5.2 Salient features of implementation

Following are the key aspects of implementation in this thesis to model the *Physical Random Access Channel* (PRACH) and corresponding random access mechanism using slotted-ALOHA as specified in the 3GPP standards.

- Access slot structure for implementing the slotted-ALOHA algorithm.
- Modifications in the existing burst structure to include the RACH preamble and the AI bursts.
- RACH transmission procedure and AICH reception procedure in the UE side physical layer RACH-handler.
- Procedures for determining collisions of preambles and sending acknowledgments in the UTRAN side physical layer RACH-handler .
- Modifications in the SDL environment, RI-handler and RxGate-handler to accommodate bidirectional, non-TTI aligned physical layer transmissions of RACH-AICH pair.

The figure 5.1 gives an overview of the implementation of RACH in URIS.

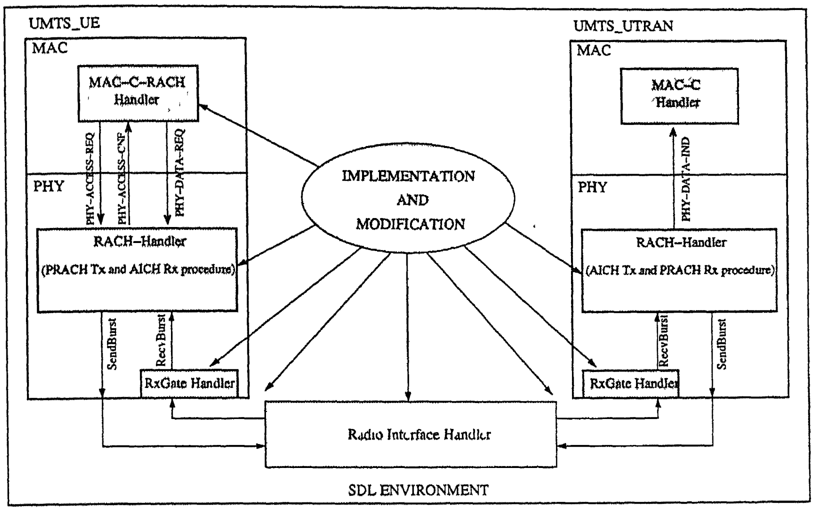



**Figure 5.1:** Overview of implementation of RACH in URIS

Each of the above points is discussed in detail below.

### 5.2.1 Access slot structure

For implementing the slotted ALOHA algorithm of the RACH an access slot structure was required. It is implemented in the physical layer RACH-handler in the base class *stUmts*. Thus the signals which mark the beginning of access slots are available to RACH-handlers in both the inherited classes *stUmts-UE* and *stUmts-UTRAN*.

The TTI signal is considered as a reference for generating access slots.

At the start of every new TTI the RI-handler sends a TTI signal to the environment of the SDL systems. A parameter indicating the length of the TTI (10 ms, 20 ms, 40 ms or 80 ms) is also passed with the signal. Since all the TTIs are time aligned the signal parameter implies all TTI lengths lower or equal to the parameter value. Figure 5.2 illustrates this.

This TTI signal from the SDL environment is passed on to the physical layer

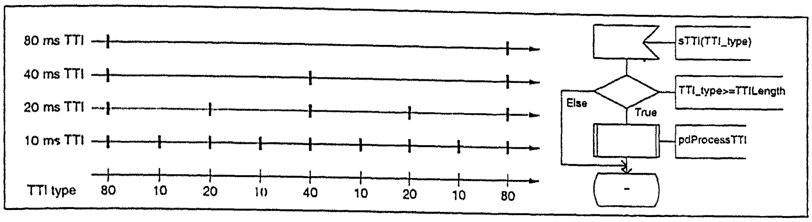



**Figure 5.2:** TTI signals with different parameters

RACH-handler via the LLME. A typical sequence of the TTIs as shown in the figure 5.2, is used in the simulator to accommodate all different possible patterns of TTI. The characteristic of the sequence is that considering a 20 ms TTI as a reference, after every 20 ms the next TTI type is a multiple of 20 ms or considering a 10 ms TTI as reference, after every 20 ms the next TTI type is 10 ms always. This scheme is very effectively used to generate access slot structure for RACH. More information about this scheme can be found in [6].

Figure 5.3 gives the SDL implementation for the access slot generation. The corresponding description is given below.

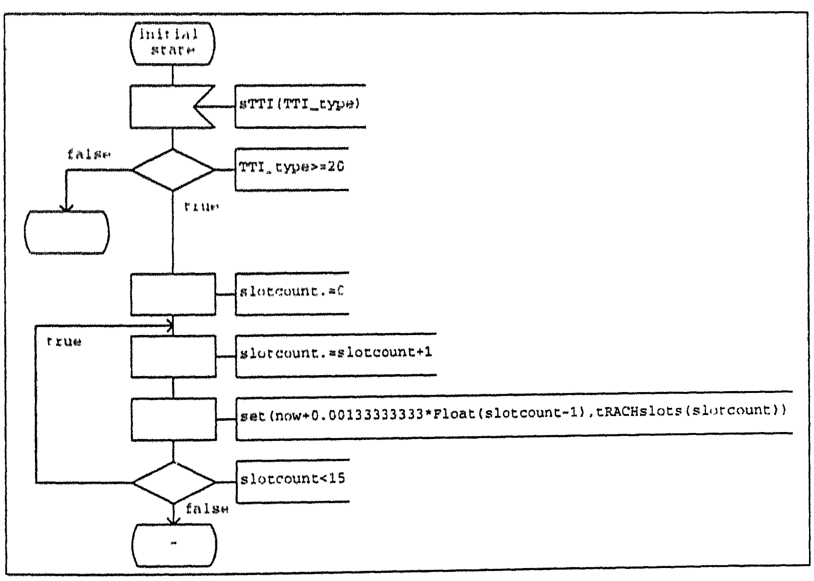



**Figure 5.3:** Implementation of access slot structure

As soon as a TTI signal is received by the physical layer RACH-handler its pa-

rameter is checked. If the parameter is 20 ms or multiple of 20 ms then the interval is divided into 15 access slots. Otherwise RACH-handler returns to the state from which it had started the procedure. A timer named *tRACHslots* with parameter *slotcount* is set every subsequent 1.3333 ms. As the parameter slotcount increments from 0 to 14, 15 timer signals are generated which mark the beginning of the access slots.

### 5.2.2 Modifications in the *burst* class

Following private data and corresponding *set* and *get* public member functions operating on these data were appended to the existing *burst* class so as to accommodate PRACH and AICH bursts which are of relevance to this thesis.

*PhysicalChType*

The simulator is currently functional only for FDD mode transmissions. Hence the concept of sending burst is represented by transmission of whole radio frames which are formed from the *Transport Block Set* (TBS) received from the MAC.

Currently bursts can only be distinguished based on transport channels, but a radio burst being a physical layer concept, bursts should be distinguished based on physical channels. Keeping this in mind an enum type called *PhysicalChType* is defined in the *uristypes* and attributed to the *burst* class. Right now only PRACH and AICH have been defined as enums of this type. Other physical channels may be defined as and when required. Corresponding *set* and *get* functions have been defined as inline functions in the header file of the *burst* class in exactly the same way as other *set* and *get* functions.

The attribute *PhysicalChType* of the burst is very effectively used in modifying RI-handler routine to transmit RACH preamble and AI burst as soon as they

are received without waiting for the start of a TTI signal. Thus these two truly physical layer bursts can be transmitted non-aligned to TTI unlike other bursts which are essentially transport channel bursts.

*PreSignature*

A preamble signature is like an ID for a preamble. Hence *PreSignature* is the most important attribute of a RACH preamble burst. It can take any integer from 1 to 16 as its value. This was absent in existing *burst* class. It is now added in the *burst* class along with its corresponding *set* and *get* inline public member functions.

*TxSlot*

For a RACH preamble, the *TxSlot* is as important attribute as the *PreSignature*. It carries the serial number of the access slot in which the preamble is transmitted.

Both *TxSlot* and *PreSignature* together are used to determine whether there is a collision between preambles transmitted by different UEs. Both these attributes are also very useful in distinguishing between AI transmission for different UEs. The use of these attributes will become more clear in sections where transmission and reception procedure of preambles and AIs in RACH-handlers on UE and UTRAN side are discussed in detail.

*AI*

*AI* which stands for Acquisition Indicator is a boolean attribute for AI bursts. AI = 1 means positive acknowledgment or ACK while AI = 0 means collision and corresponding negative acknowledgment or NACK. Thus AI with corresponding *set* and *get* inline public member functions is the most important attribute of the AI burst which is used to transmit acknowledgment of RACH preambles.

The access slot structure and modifications in burst class prepared the required base for further development of the RACH model.

### 5.2.3 RACH procedures on UE side

The following description is with reference to figure 5.4 which shows the algorithms implemented in *Specification Description Language* (SDL) within the RACH-handler on the UE side.
At the end of the startup routines UE side RACH-handler comes in state *Started.*

As soon as it receives the primitive *sPHY-Access-Req* from the MAC layer preamble burst attributes viz power, signature and access slot are randomly selected from a uniform distribution. Other useful parameters like preamble retransmission counter and preamble power ramping step size are initialised. Currently these values are taken from *config.ini* file but in the near feature these values will come from RRC layer as mentioned in the specifications.

The RACH-handler than goes into the state *WaitSlotBoundry* in which it waits for the timer signal corresponding to the selected access slot to arrive.

As soon as it receives the timer signal *tRACHslots* corresponding to the selected access slot the RACH-handler builds up a preamble burst structure assigning various necessary attributes to the burst by using the corresponding *set* functions. It then sends the preamble burst and sets a timer *tTauPP* for exactly three access slots and returns to the state *Started.* This timer is used to timeout the waiting period for acknowledgments (AI) and start the procedure for retransmission of preamble at higher power.

Suppose the preamble power is not sufficient as a result of which neither positive nor negative acknowledgment is sent from UTRAN. Consequently the timer

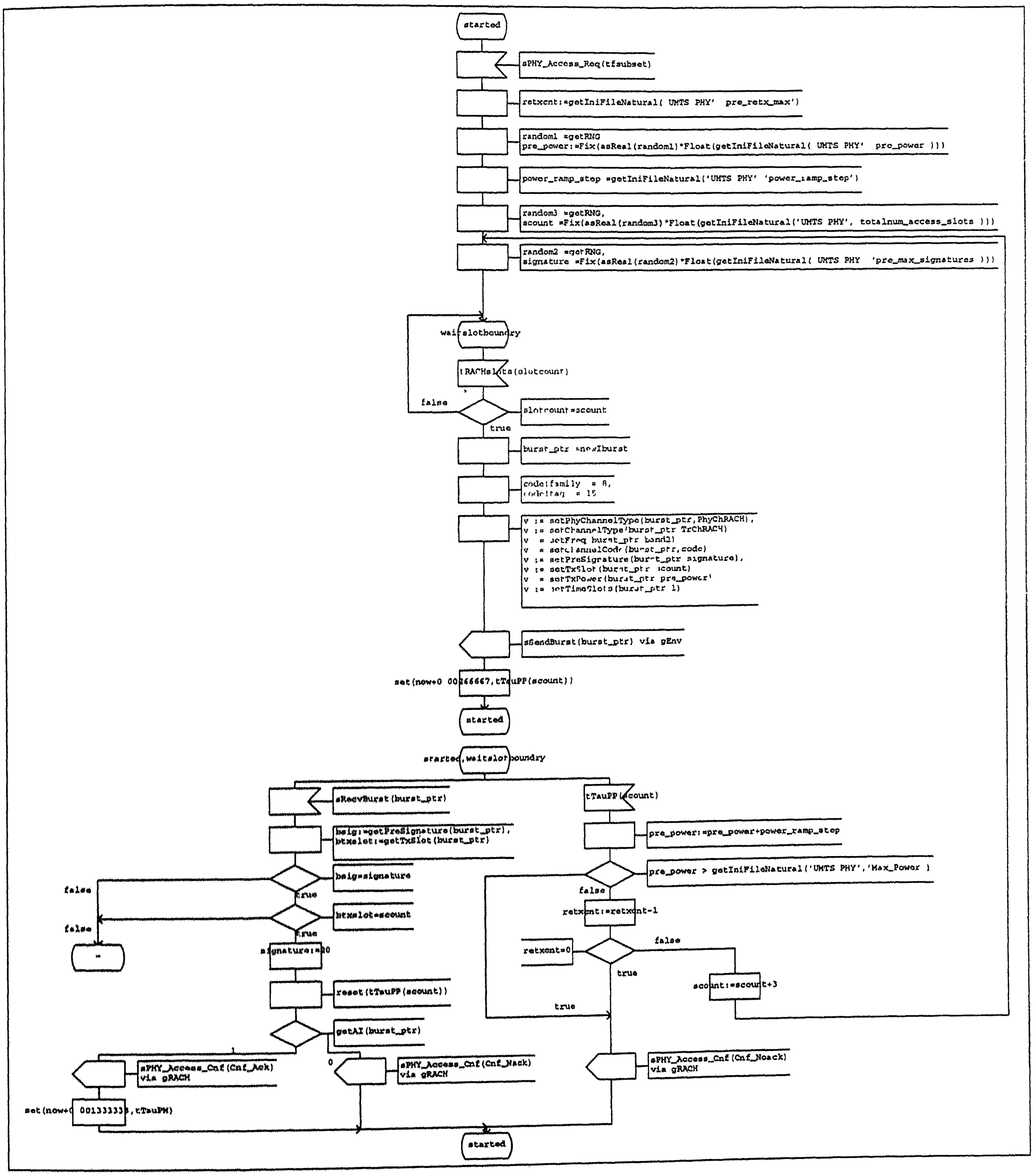



**Figure 5.4:** RACH procedures on UE side

*tTauPP* expires. This triggers the preamble retransmission procedure in which the power is first increased. If the preamble power after incrementing is less than the maximum allowable preamble power then the preamble retransmission counter is decremented. Again if the count is not zero preamble is retransmitted with newly selected signature and access slot. If either the preamble power exceeds the maximum allowable or the retransmission counter gets exhausted the primitive *sPHY-Access-Conf* is sent to MAC with a parameter indicating that *No Acknowledgement* (NoACK) was received.

Consider the other possibility when the preamble power is sufficiently high. The RACH handler receives an AI in form of the signal *sRecvBurst* before the timer *tTauPP* expires. A particular UE's RACH-handler can receive this signal even as a response to some other UE's preamble transmission. Hence, when this signal is received, it becomes necessary to distinguish between different UE's. Now, AIs are sent in response to receipt of preamble having particular signature and not preamble from a particular UE. The attributes *PreSignature* and *TxSlot* are used to a good effect in distinguishing UEs here. An AI is considered as a response to a particular UE's preamble only if the received burst has the same signature as that of the sent preamble. Furthermore, a comparison of *TxSlot* ensures that an AI which is a response to the same signature as the given preamble's but to the preamble transmitted in a different access slot than the given preamble is not accepted. Thus a two step comparison process ensures that only the valid AI responses are accepted by the RACH-handler of a particular UE and processed further.

As soon as the valid AI is accepted the timer *tTauPP* is reset and AI value is tested for the type of acknowledgment. A negative acknowledgment simply prompts the passing of *sPHY-Access-Conf* with NACK info to MAC. While a positive acknowledgment implies that RACH message part transmission has to

follow after a fixed time of 3 or 4 access slots. Hence *sPHY-Access-Conf* with ACK info is passed to MAC after setting a timer *tTauPM.*

Irrespective of the nature of acknowledgment that is passed to the MAC the RACH-handler ends up in the state *Started.*

Mapping of the transport channel RACH's TBS to PRACH message part and subsequent transmission of the message part was already implemented in the RACH-handler.

### 5.2.4 RACH procedures on UTRAN side

The following description is with reference to figure 5.5 which shows algrithms implemented in SDL in the RACH handler on the UTRAN side.

At the end of the startup routines the UTRAN side RACH handler is also in state *started.*

Since the same signal *sRecvBurst* is used to receive both the RACH message and preambles these two types of bursts must be distinguished based on some common attribute which has different values for these two types of burst. For this purpose an attribute named *TimeSlots* is used. In the simulator so far for all the transport channel bursts *TimeSlot* takes a default value of 15 because of the whole radio frame transmissions in FDD mode. But for RACH preambles *TimeSlot* is given an approximate value of 1 because RACH preamble burst spans 4096 chips, which is almost equal to 1 timeslot (not the access slot).

Thus after the receipt of *sRecvBurst* signal with burst as its parameter, number of timeslots is extracted from the burst structure using the *getTimeSlots* member function. If this number turns out to be one, following procedure is used for

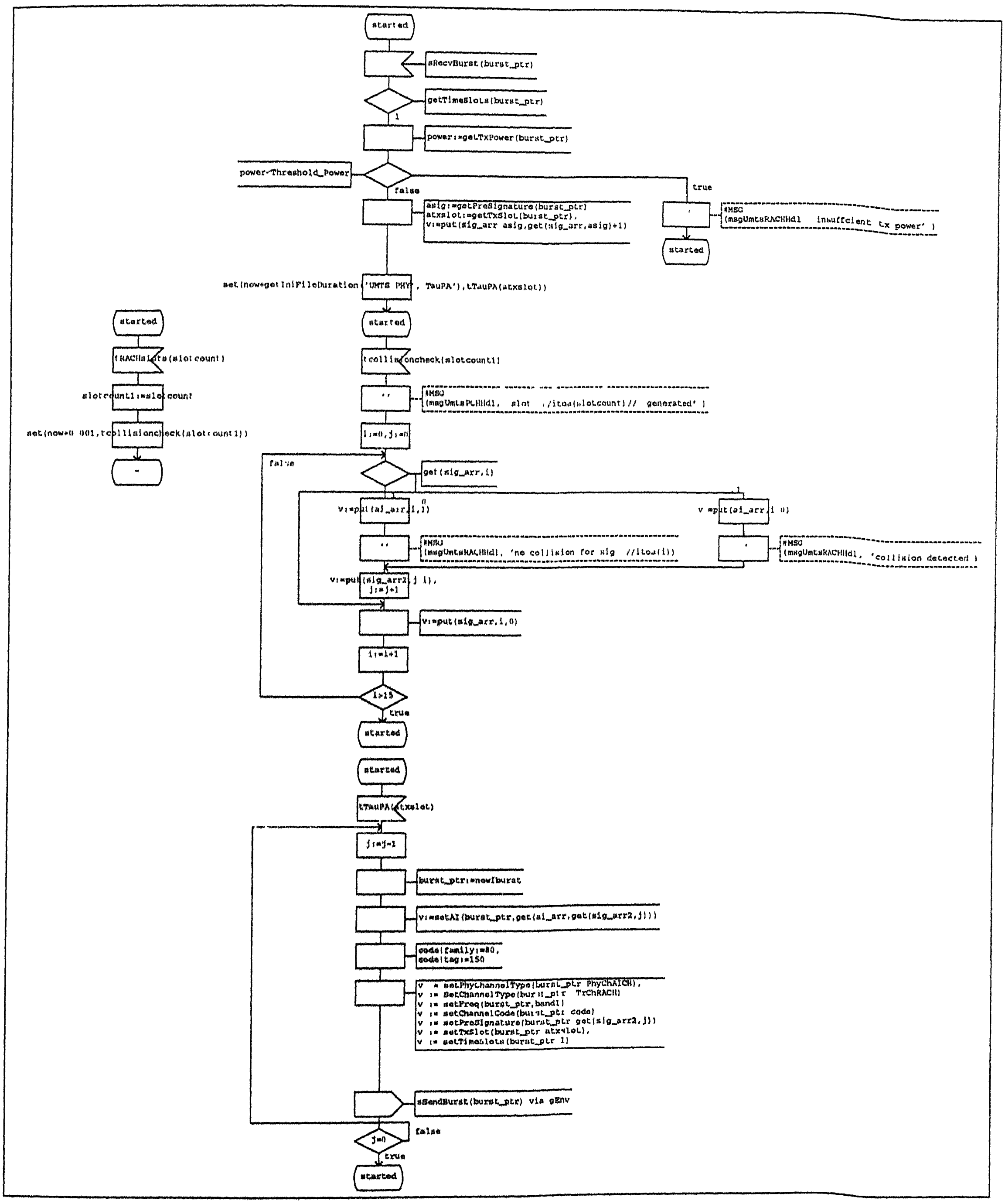



**Figure 5.5:** RACH procedures on UTRAN side

reception and acknowledgment of RACH preambles.

First of all the preamble power is measured. If it is less than the threshold power the RACH-handler immediately returns to the state *Started* without any further actions. Such a response of the RACH-handler causes its counterpart on UE side to timeout and retransmit the preamble with higher power and different signature and also in different access slot.

Whenever the preamble power exceeds the threshold, first of all the values of the two most important attributes of the preamble, namely *PreSignature* and *TxSlot* are extracted. An array *sig-arr* of natural numbers of size 16 where each index corresponds to a signature is declared. After getting the value of *PreSignature* from the received burst, the value of the array element indexed by the signature value is incremented by one ( starting from zero ) A timer *tTauPA* which determines the fixed time of transmission of acknowledgment after the reception of the preamble is set and the RACH-handler returns to the state *Started.*

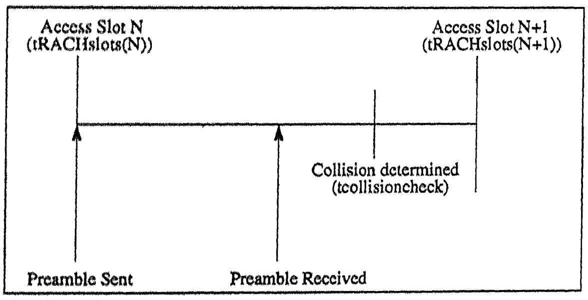



**Figure 5.6:** Timings at the UTRAN side

As shown in the figure 5.6 the signal, *sRecvBurst* carrying preamble burst is received by the handler exactly in between two consecutive timer signals *tRACHslots*. First of this marks the start of the access slot in which the preamble was transmitted. Thus as soon as the timer signal *tcollisioncheck* after the signal *sRecvBurst* with RACH preamble is received all the elements of *sig-arr* are

checked.

If within the duration of one access slot any of the elements of *sig-arr* has a value greater than one, the signature, (the preamble with the signature) which indexes that element, is considered to be in collision. Another array named *ai-arr* of natural numbers of size 16 is also declared, elements of which are the AI values. For each signature index of *sig-arr* which has element value one, there exists an element of value 1(AI = +1) for the same index ( signature ) in the *ai-arr*. Whereas a value greater than one in the *sig-arr* corresponds to a 0 (AI = -1) in the *ai-arr*. There is another array called *sig-arr1* which stores the signatures being used during a particular access slot, i.e.those which are either successful or face a collision.

After determining which signatures have faced a collision, the RACH-handler returns to the state*Started*. Shortly after this the timer *tTauPA* expires. On receipt of the timer signal *tTauPA* all the signatures used during a particular access slot are acknowledged through an AICH burst where the attribute *AI* indicates a positive or negative acknowledgment. The arrays, *ai-arr* and *sig-arr1* are used in combination to set value of *AI* for forming the AI burst structure.

In summary the distinguishing of preamble from message part, checking for sufficient transmit power, determining of collision and sending of acknowledgment are the tasks performed in the UTRAN side RACH-handler.

### 5.2.5 Other miscellaneous modifications

RACH as such is a unidirectional channel but in physical layer during the preamble phase PRACH is also supposed to send and receive acknowledgments. It is thus a bidirectional channel in physical layer from the implementation point of view.

RI-handler of the simulator was capable of handling only TTI-aligned transmis-

sions of the normal transport channel burst as described previously.

To eliminate the above mentioned shortcomings from the existing version of the simulator changes were made in:

- The C++ routines which model SDL environment and RI-handler
- The SDL structure, namely RxGate-Handler, which acts like a gateway between SDL environment and SDL system

These miscellaneous modifications are described below.

### SDL environment

Since RACH is not a dedicated channel uplink transmissions were targetted to all the base stations available in the simulator.

With the modifications made, RACH transmissions are now bidirectional. Direction of transmission is determined from the value of attribute *Freq* of the burst structure because uplink and downlink frequencies are different in FDD mode. Thus uplink transmissions are as before, broadcast to all the base stations, whereas the downlink transmissions are similarly broadcast to all the mobile stations. This is achieved by appropriately specifying the contents of receiver array in *env.cpp*.
This change was a must to accommodate transmission of preamble acknowledgments using AIs.

### RI-Handler

Major functions of the RI-handler and its inter-working with the SDL environment and other C++ modules like the channel model is shown in figure 5.7.
Bursts from the SDL environment reach the RI-handler via the global *evTraffic* event. These bursts are subsequently added to a list of bursts. If the burst has exactly one dedicated receiver, it is just added to the end of the single linked list.

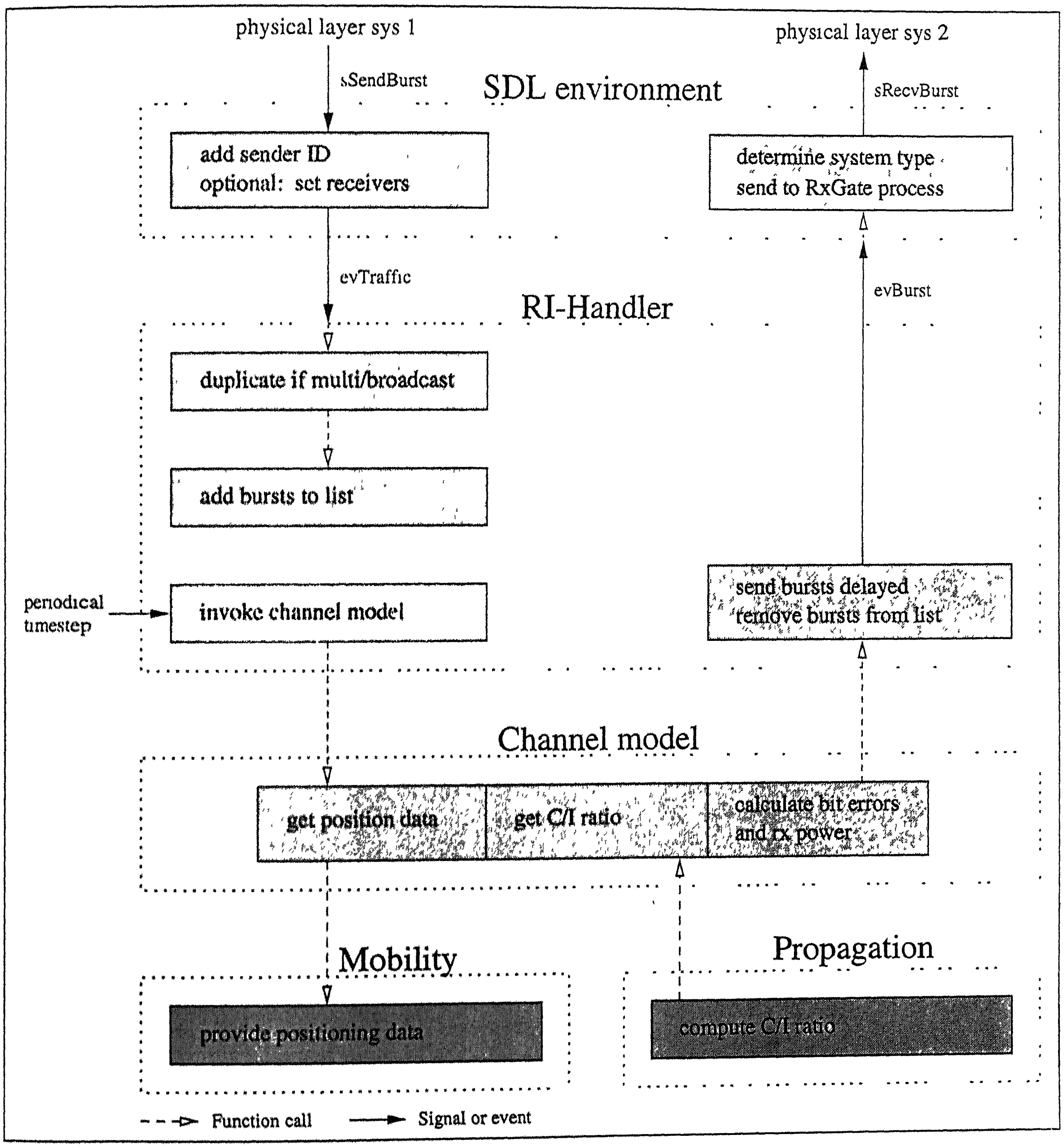



**Figure 5.7:** RI-handler and its relationship with other modules in C++

But if it has more than one or no dedicated receiver at all, the burst is meant to be multicasted or broadcasted. Hence the burst is duplicated as often as needed and every copy gets one dedicated receiver. These copies are then appended to the list.

RI-handler sends an *evTimeStep* signal to itself every timeslot, i.e.10/15 ms for periodic evaluation of the bursts. On receipt of this signal the entire linked list of bursts is passed on to the channel model which drops certain bursts, based on the statistical model of channel. The bursts that are still in the list are delivered to the SDL environment via the *evBurst* event. For the FDD mode even though the signal *evTimeStep* comes every time slot, bursts are not passed to the channel model and subsequently to the SDL environment until a TTI signal arrives. Thus effectively not the bursts, but the entire 10 ms radio frame is transmitted simultaneously on the occurrence of TTI signal.
r Above scheme of RI-handler is not acceptable for the RACH preamble transmission which is not aligned to the TTI.The RACH message part which follows a successful RACH preamble after 3 or 4 access slot also loses any kind of alignment with TTI. This problem is tackled as described below.

As soon as the event *evTraffic* is received, its physical channel type and transport channel type attributes are extracted. If the burst belongs to the physical channel types PRACH or AICH,i.e.the burst is a RACH preamble or an AI, it is not added to the link list. It is simply duplicated as many times as the number of receivers and sent to the SDL environment in the signal *evBurst.*
Furthermore, if the burst belongs to transport channel RACH i.e.the burst is a RACH message, it is added to the linked list where operations similar to any other transport channel burst may take place. But unlike for other transport channels, the RI-handler does not wait for arrival of a TTI signal. Instead the RACH message burst is passed to the channel model and transmitted using *evBurst* to the SDL environment instantly after duplication and addition to the linked list.

The above mentioned modification in the RI-handler is one of the most critical parts of the thesis since without this it was impossible to pass the RACH bursts between physical layer RACH-handlers on UE and UTRAN side without any dependence on TTI.

### RxGate Handler

RxGate handler has a RxSwitch procedure that filters out bursts from various transport channels and passes them on to the respective physical layer handlers via the respective gates. Since RACH is supposed to be only an uplink channel there was no filter for RACH bursts in the RxSwitch on the UE side.

This limitation in the RxSwitch on the UE side was easily eliminated by implementing constructs in SDL for the transport channel RACH similar to other downlink transport channels.

CHAPTER 6

# Simulations

This chapter starts with a brief description of the classical slotted ALOHA algorithm which forms the basis for this thesis It is followed by a summary of the research done so far specifically on the *Random Access Channel* (RACH) in UMTS that has served as a guideline and a motivation for the thesis.

## 6.1 Slotted ALOHA

The history of random access of a shared radio communication channel can be traced back to 1970s when for the first time Norman Abramson and his colleagues developed a protocol called ALOHA. ALOHA is a truly *Transmit at Will* kind of protocol. In this protocol each node transmits a new packet as soon as it is generated. Assuming constant packet lengths, if there is a packet transmission within the duration of one packet length before or after the transmission of a particular packet there would be an interference between these transmissions and none of the packets will be received correctly at the receiver. This event is termed as a *collision*. Once the event of a collision is known to transmitters, each packet involved in a collision is retransmitted after a random delay so that the same packets may not collide once more.

Consider the combined load generated by new and collided packets to be Poisson distributed with a mean of **G** packets/slot (slots are equal to packet lengths).

$$P[k\ arrivals\ in\ time\ T=1] = \frac{G^k e^{-G}}{k!}$$

Now, the probability of a successful transmission is given by:

$$P_s = P[k=0\ arrivals\ in\ time\ T=2\ slots] = \frac{(2G)^0 * e^{-2G}}{0!} = e^{-2G}$$

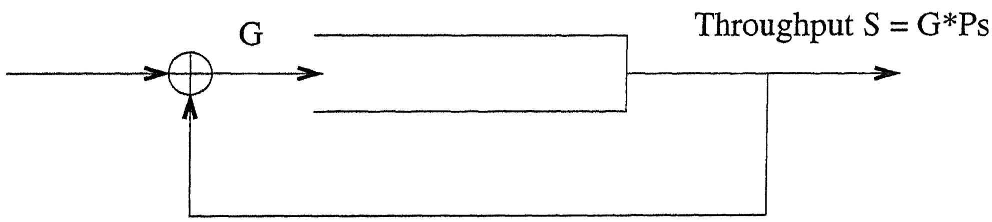



**Figure 6.1:** ALOHA channel as a queue with feedback

Hence,

$$Throughput\ S = Ge^{-2G}$$

The maximum throughput occurs at $G = 1/2$ with the maximum value being $S = 0.174$.

Thus pure ALOHA is very wasteful of resources and has a very poor capacity.

If all the nodes are synchronized, instead of allowing the nodes to transmit new packets at arbitrary times (Free Will), they may be constrained to transmit only at the beginning of well-defined slots. This is referred to as *slotted ALOHA*. This causes a vast improvement in capacity of a shared channel since the vulnerable period for collisions is reduced to just one packet duration.
Again considering the Poisson assumption and referring to the figure 6.1 above, throughput of the slotted ALOHA system is given by:

$$Throughput\ S = Ge^{-G}$$

In this case the maximum throughput occurs at $G = 1$ with the maximum value being $S = 0.368$.
Thus with slotted ALOHA the capacity of the shared channel is directly doubled as compared to pure ALOHA. A detailed treatment to slotted ALOHA and other classical random access schemes is given in [2] and [24]. A different approach to analyse the throughput of slotted random access schemes is presented in [25].

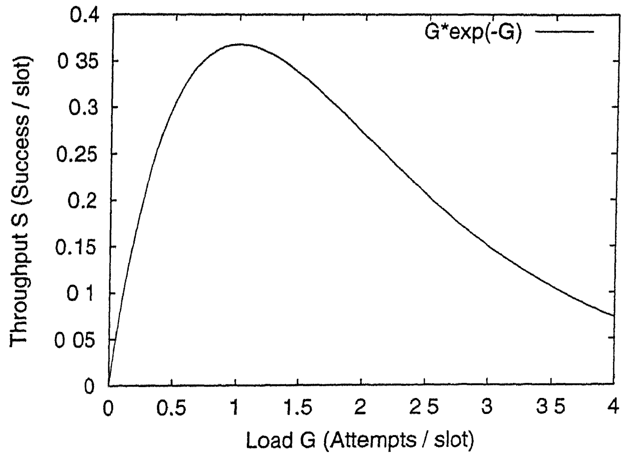



**Figure 6.2:** Throughput vs load characteristics of classical slotted ALOHA

## 6.2 Slotted ALOHA and RACH in UMTS

Statistically the RACH is UMTS is much the same as the classical slotted ALOHA algorithm described above with two major differences:

1. The packets can be distinguished by one of the sixteen signatures.

   Thus there is sixteen fold increase in capacity but there is sixteen fold increase in resources too. Hence, on normalisation, use of sixteen signatures to distinguish packets does not effect any improvement in capacity as such.
2. The actual packet is transmitted only after getting an acknowledgment for a short preamble packet.

   The preamble may be thought of as a sniffer to judge the channel conditions so that the actual packet has a much higher probability of success.

Considering only the first aspect, a highly simplified model of the RACH may be thought of for the sake of a simplified analysis as follows.

Each UE can transmit its message at the beginning of well-defined slot boundaries just like classical slotted ALOHA does. But unlike the simple slotted ALOHA the UE randomly selects one of the available signatures before transmitting. Each of the signatures can be chosen with equal probability.

Corresponding *Throughput-Load* characteristics are shown in figure 6.4. These curves serve as a good reference for comparing the result of simulation with different number of signatures and thereby validate the implementation in this thesis.

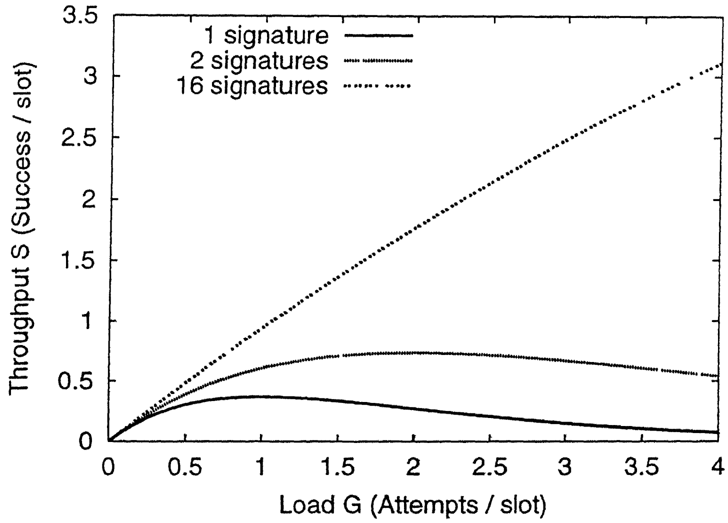



**Figure 6.4:** Throughput Vs Load characteristics of slotted ALOHA with different number of signatures

## 6.3 State of Research on RACH in UMTS

Ever since the formulative stages of specifications for the UMTS, research has been going on to evaluate throughput and delay performance of the slotted ALOHA algorithm in WCDMA environment and especially for the RACH as specified in 3GPP standards. The research so far on *RACH in UMTS* has not been collateral. It has progressed mainly in the following two directions:

1. Dealing with issues like power control, interference minimization and signature allocation which are typical to the WCDMA environment and evaluating the performance of RACH with respect to these parameters. [20], [21], [22].

2. Evaluating a purely statistical model of the RACH. The power control and interference issues are neglected altogether. [24], [25].

This thesis seeks its motivation from the second line of research because the *UMTS Radio Interface Simulator* (URIS) is in a developmental stage without a properly developed channel model for WCDMA. The thesis also paves the way for a more holistic research considering both the issues simultaneously once the URIS is fully functional.

### 6.3.1 Performance results of the RACH from the research so far

[20] describes the performance of two candidate schemes for RACH transmission namely *Message Power Ramping* and *Preamble Power Ramping* while the 3GPP specification for RACH were being formulated. It was shown by simulations that preamble power ramping outperformed other alternatives for controlling the random access power in WCDMA. The scheme was also found to be highly robust in different scenarios.

In [21] a different enhanced power ramping scheme with the use of multi-threshold detection in receiver was suggested. It claimed to increase the throughput of the access channel and also reduce the interference level. Moreover the multi-threshold detection algorithm allowed the base station to send a fast acknowledgment back to the mobile reducing the access delay thereby.

The research work in [22] proposed a modified version of the preamble power ramping scheme. In the original scheme the message part uses the same signature that was used by the last, successfully transmitted preamble. But in the proposed scheme a new, random selection of signature for the message part was suggested. The simulation results showed a significant improvement in the throughput of RACH.

All the above work was mostly of simulative nature and with a focus on suggesting power control and code allocation schemes that achieved an improvement in throughput, coupled with reduction in interference and delay

Reference [24] discusses the stability issues in slotted ALOHA algorithm and suggests an *Adaptive Dynamic Persistence Algorithm* wherein the persistence value used in MAC RACH procedure is made to adapt to changing load conditions. This works in such a way that the slotted ALOHA algorithm operates in the stable region. It also studies the improvement in throughput caused due to capture effect at the receiver. Capture effect referes to a special ability of the receiver to distinguish certain high powered transmissions from a set of interfering transmission during the event of a collision.

RACH performance based on the MAC layer resource allocation schemes to different access service classes is discussed in [23]. In the non-overlapping scheme each ASC is assigned a separate set of MAC resources while in the other, overlapping scheme each ASC uses resources from two or more sets of MAC resources. By simulations, the overlapping scheme was found to be difficult to design but proved to be more efficient than its counterpart which is easier to design but less efficient.

Some analytical treatment, very specific to the current RACH standard, can be found in [25]. The analysis and simulations are made with a simplified model of the RACH without considering preamble power ramping. It is propounded that the RACH, as specified in UMTS, has a significantly higher capacity than the classical slotted ALOHA. The statistical model of RACH developed in this paper and its analysis for throughput is discussed below.

- It is assumed that the combined traffic due to the newly generated preambles and the retranmissions is poisson with rate $G/1.33$ transmissions/ms.
- As shown in figure 6.5 the total length of the preamble, preamble to message

gap and the message transmission time combined is denoted by $T_m$

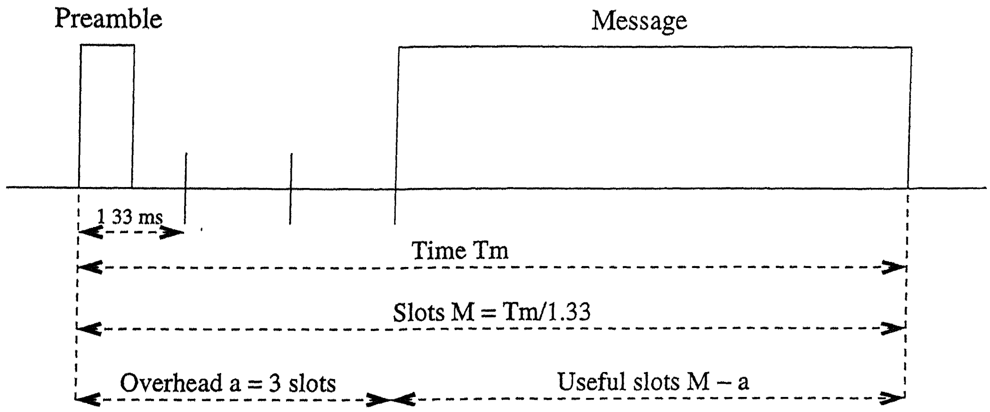



**Figure 6.5:** Pictorial discription of period $T_m$

- The evolution of protocol in time is represented as a sequence of idle and busy periods as shown in figure 6.6

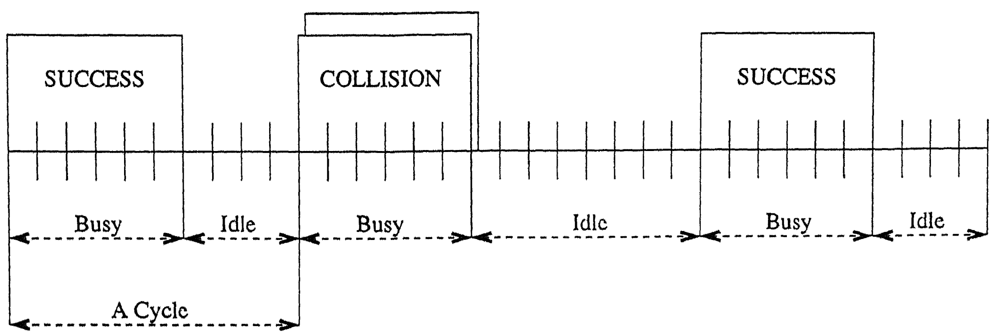



**Figure 6.6:** Time axis divided into idle and busy periods

- Further, it is assumed that each busy period resulting in either a success or a collision is of the same length $M$ slots.
- Now, a cycle is defined as the combination of a busy period followed by an idle peroid. Then the average throughput is defined as

$S = \frac{E[Useful]}{E[Idle] + E[Busy]}$ wherein,

$$E[Idle] = \frac{e^{-G}}{1 - e^{-G}}$$

$$E[Busy] = M$$

$$E[Useful] = P[Success](M - a)$$

Thus, throughput

$$S = \frac{P[Success](M - a)(1 - e^{-G})}{e^{-G} + M(1 - e^{-G})}$$

- The authors then derive the probability of success for the case when two or more preambles overlap and more than one signature is used as follows

$$P_n[Successful\ cycle] = \sum_{k=1}^{\infty} \frac{G^k}{k!} \frac{e^{-G}}{1 - e^{-G}} \sum_{m=0}^{n-1} \sum_{j=0}^{min(n-m,k)} \frac{j}{n - m} X$$

$X\ P[j\ signnature\ with\ 1\ preamble,\ m\ signature\ empty](n,\ k)$

- From the analysis, normalised throughput as a function of attempt rate G is obtained.
- The RACH capacity is found to be significantly higher than the classical slotted ALOHA algorithm.
- Improvement in capacity is attributed to the preamble phase of RACH transmissions. It is explained that the preamble phase does some sort of carrier sensing and thereby causes a significant improvement in capacity.

During the course of the thesis, the analysis discussed above was found to have some serious drawbacks. The corresponding results were also found to be arguable. The drawbacks and the suggestions to improve upon them are presented below.

- The assumption, that busy periods and idle periods are of same fixed lengths M suffers from following drawbacks.
  1. It is well known in such analysis that a busy period or an idle period cannot be of a fixed length $M$ slots. It is a random variable.
  2. Busy period lengths in events of success and collision cannot be same.

- The probability mass function of the busy period can instead be found as follows:

$P[Busy\ period\ =\ 1\ slot]$

$= P[There\ is\ no\ arrival\ at\ the\ next\ slot\ boundry\ and\ an\ idle\ period\ starts]$

$= e^{-G}$

Extending this argument for k slots gives the p.m.f. of busy period as below:

$P[Busy\ period\ =\ k\ slots]$

$= P[There\ is\ one\ or\ more\ arrival\ at\ k-1\ slot\ boundries\ and\ no\ arrival\ at\ the\ k^{th}]$

$= (1\ -\ e^{-G})^{k-1}\ e^{-G}$

The average busy period can then be easily determined as $\frac{1}{e^{-G}}$

- Similarly the p.m.f. of idle period can be found out to be:

$P[Idle\ period\ =\ k\ slots]$

$= P[There\ is\ no\ arrival\ at\ k-1\ slot\ boundries\ and\ one\ or\ more\ arrival\ at\ the\ k^{th}]$

$= (e^{-G})^{k-1}\ (1\ -\ e^{-G})$

Hence the average idle period can be calculated as $\frac{1}{1\ -e^{-G}}$ and not $\frac{e^{-G}}{1\ -e^{-G}}$ as derived in the paper.

- The authors attribute the improvement in capacity to the preamble phase, but the preambles are overhead in terms of statistical throughput of RACH They may improve the overall capacity of UMTS system by minimising the *Multiple Access Interference* (MAI), but that is not possible to observe from the analysis in the paper since it does not take into the account the effect of WCDMA channel.

Thus for the RACH model in the thesis, results obtained by considering RACH as a multi-signatured slotted ALOHA system are enough for the comparative study of the simulative results.

The analysis in the paper discussed above may be improved as per the suggestions given in the thesis and used for comparision with more complex model of RACH.

## 6.4 Simulations in the thesis

Simulations were carried out with Poisson traffic source. The following methodology was followed to perform simulations.
Each UE receives a request for random access in the MAC layer periodically such that the inter-arrival times of the requests are exponentially distributed. Thus the requests at the MAC layers come from a Poisson process. Furthermore, a linear combination of Poisson sources is also Poisson. Hence, the overall traffic from all the UEs is also Poisson.
Although Poisson traffic is quite unrealistic it is used for simulations so that the results can be easily compared with those that are already available from analysis of classical slotted ALOHA or other means.

Since a precise channel model for WCDMA was not available, simulations with power ramping would have been highly unrealistic. Hence, preamble power ramping phenomenon was not simulated, although the requist functionality was modelled in the physical layer RACH-handlers. Thus, just one preamble was transmitted and once it was acknowledged positively the message part was transmitted. The message part transmission is supposed to use the same signature as successful preamble. Moreover, it is also transmitted, a fixed time after the transmission of the successful preamble. Therefor it is assumed in the simulations that a successful preamble transmissions implies a successful message part transmission. Such an assumption is highly unrealistic but it provides a very good means for validating the implementation. Since the scenario is much the same as classical slotted ALOHA, simulations using one preamble signature should yield results similar to classical slotted ALOHA.
For a further validation, similar simulations were also carried out for two and sixteen signatures and their results compared with the multisignatured slotted ALOHA analysed earlier.

For a constant small enough value of inter-arrival time (0.05 seconds) the number of UEs was varied from 5 to 150.

The most important parameters for performance evaluation of the RACH are *Throughput* and *Delay.* Following data was collected by using SDL probes for measurement of these parameters.

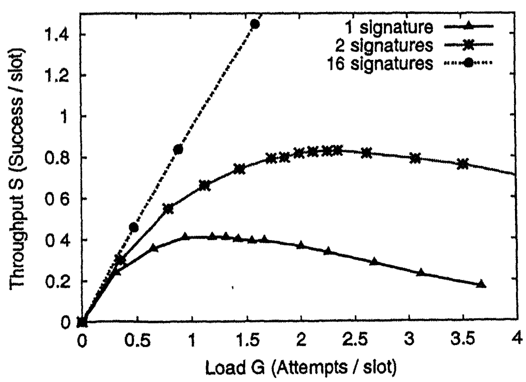
1. Total number of access slots generated
2. Number of access attempts made at physical layer
3. Number of access attempts made at MAC layer
4. Number of successful transmissions
5. Number of collisions
6. MAC layer transmission delay

Following basic set of equations were used to calculate *load* and *throughput.*

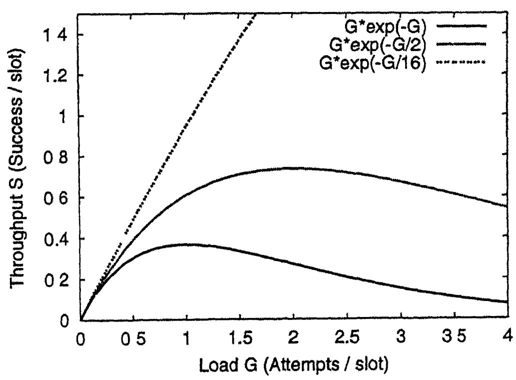
$$Load\ G = \frac{Number\ of\ attempts\ made}{Total\ number\ of\ access\ slots\ available}$$

$$Throughput\ S = \frac{Number\ of\ successful\ transmissions}{Total\ number\ of\ access\ slots\ available}$$

Throughput-load characteristics as shown in figure 6.4 were obtained.



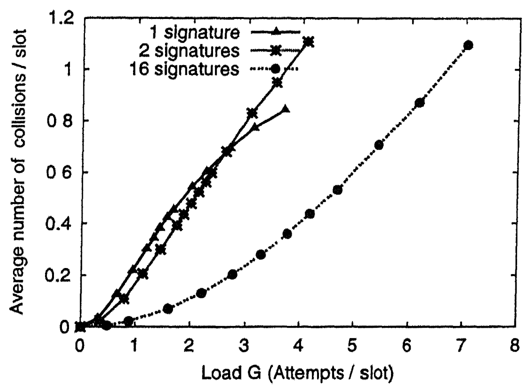

Throughput vs. load characteristics of RACH by simulations



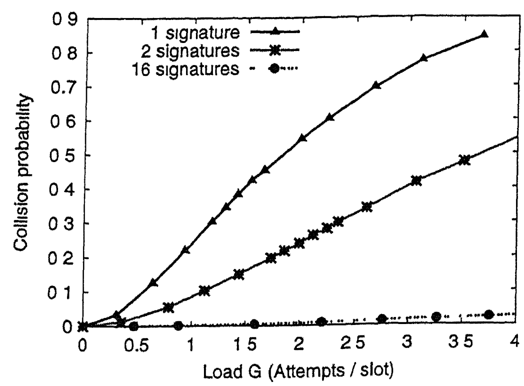

Throughput vs. load characteristics of slotted ALOHA

A comparison of the curves from the simulation results with the ones derived from analysis of multi-signature slotted ALOHA, clearly validates the RACH implementations in this thesis.

Some more results are presented and discussed below.




Average number of collisions vs. load




Collision probability vs. load

It can be seen that the average number of collisions per slot increases monotonically with offered load. This is quite obvious.

Also the average number of collisions per slot decreseas as the number of signatures is increased. This is as expected. But the crossing of the curves for 1 signature case and 2 signature case is a little bit misleading.
With one signature there can be at the most one collision per slot. Hence, the curve tends to saturate at the value 1; whereas with two signature, each of the two signature can face collisions and hence the average number of collisions can be more than 1 and the curves overlapp.

To avoid this confusion, the average number of collisions is further normalised with respect to number of signatures. In effect the collision probability is plotted against the offerd load. The results are now much more tangible and in line with the expections.

Following curves show the mean delay performance with respect to offered load.

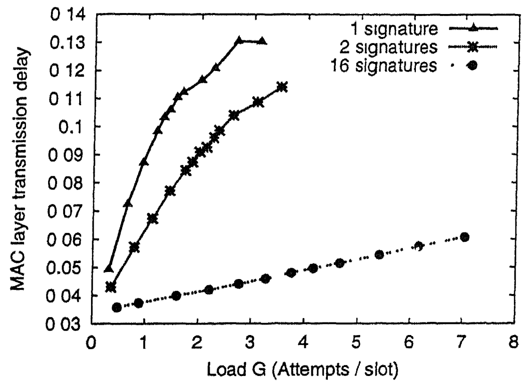



**Figure 6.7:** Delay vs.load

Again, the results are as expected. With an increase in offered traffic the mean tranmission delay as seen from the MAC layer increases monotonically. The transmission delay sums up the actual message part length and the time taken for preamble phase. If the preamble is positively acknowledged, the delay is slightly more than the message part legth because of the preamble phase. 20 ms message part is used in the simulations. The minimum values for the delay are found to be slightly greater than 20 ms. Moreover, if the preamble is negatively acknowledged due to a collision, the backoff times, which are in the multiple of 10 ms, also add up. In such a case, the delay would be slightly greater than some multiple of 10 ms.

CHAPTER 7

# Conclusions

## 7.1 Results

Following are the two most noteworthy outcomes of the thesis.

1. Development of the physical layer model for the *Random Access Channel* (RACH) in *UMTS Radio Interface Simulator* (URIS)
2. Performance evaluation of the *Random Access Channel* (RACH) in UMTS

### 7.1.1 Development for the physical layer model of RACH in URIS

A model of the RACH in the physical layer confirming quite closely to the latest 3GPP specifications was developed and integrated into the existing structure of URIS.

The current, enhanced version of URIS has thus become more realistic. In the earlier version all UEs issueing a request for a dedicated connection were granted one without any contention phase. With the model of the RACH developed in the thesis only those UEs which are able to get through the initial contention phase will be granted a dedicated resource.

A simplified model for slotted ALOHA with multiple signatures was developed and analysed for throughput.

The slotted ALOHA algorithm for the RACH implemented in this thesis was validated by performing simulations with Poisson traffic for 1, 2 and 16 signatures. The load-throughput characteristics obtained by simulation results were

compared with the ones obtained by simplified analysis.

### 7.1.2 Performance evaluation of RACH in UMTS

Throughput behaviour of RACH with changing Poisson traffic was found to be very similar to the classical slotted ALOHA.

Collision probability increases with increasing monotonically with offered load and decreases with increasing preamble signature.

MAC layer transmission delay also increases monotonically with offered load.

## 7.2 Outlook

More realistic RACH simulations with preamble power ramping and multiple ASC can be performed once a channel model for URIS is fully developed.
This would further lead to a more exact and precise performance evaluation of the RACH.

Once the HTTP traffic is available, simulations on the RACH can be performed with more realistic load scenarios and its performance evaluations can be carried out.

The modeled RACH can be used in combination with other downlink channels to simulate asymmetric traffic scenarios which will be a typical application in UMTS in the near future.

The implementations in the thesis can serve as a guideline for modeling *Physical Common Packet Channel* (PCPCH) which is quite similar to the *Physical Random Access Channel* (PRACH).

## LIST OF ABBREVIATIONS

| | |
|---|---|
| 3G | 3rd Generation |
| 3GPP | 3rd Generation Partnership Project |

**A**

| | |
|---|---|
| ACK | Acknowledgement |
| ADT | Abstract Data Type |
| AI | Acquisition Indicator |
| AICH | Acquisition Indicator Channel |
| AM | Acknowledged Mode |
| AS | Access Stratum |
| ASC | Access Service Class |

**B**

| | |
|---|---|
| BCCH | Broadcast Control Channel |
| BCH | Broadcast Channel |
| BMC | Broadcast/Multicast Control |
| BS | Base Station |

**C**

| | |
|---|---|
| CB | Cell Broadcast |
| CBR | Constant Bit Rate |
| CCPCH | Common Control Physical Channel |
| CCCH | Common Control Channel |
| CCTrCH | Coded Composite Transport Channel |
| CDMA | Code Division Multiple Access |
| CEPId | Connection Endpoint Identifier |
| CN | Core Network |
| CPCH | Common Packet Channel |
| CPICH | Common Pilot Channel |
| CRC | Cyclic Redundancy Check |
| CSICH | CPCH Status Indicator Channel |
| CTCH | Common Traffic Channel |

**D**

| | |
|---|---|
| DCCH | Dedicated Control Channel |
| DCH | Dedicated Channel |
| DC-SAP | Dedicated Control SAP |
| DL | Downlink |
| DLL | Data Link Layer |
| DPCCH | Dedicated Physical Control Channel |
| DPCH | Dedicated Physical Channel |

| | |
|---|---|
| DPDCH | Dedicated Physical Data Channel |
| DS-CDMA | Direct-Sequence Code Division Multiple Access |
| DSCH | Downlink Shared Channel |
| DTCH | Dedicated Traffic Channel |
| DTX | Discontinuos Transmission |

**E**

| | |
|---|---|
| ETSI | European Telecommunication Standards Institute |

**F**

| | |
|---|---|
| FACH | Forward Access Channel |
| FAUSCH | Fast Uplink Signalling Channel |
| FCFS | First Come First Served |
| FDD | Frequency Division Duplex |

**G**

| | |
|---|---|
| GC-SAP | General Control SAP |
| GPRS | General Packet Radio Service |
| GSM | Global System for Mobile Communication |

**H**

| | |
|---|---|
| HTTP | Hypertext Transfer Protocol |

**I**

| | |
|---|---|
| ICI | Interface Control Information |
| ID | Identifier |
| IDU | Interface Data Unit |
| IMEI | International Mobile Equipment Identity |
| IMSI | International Mobile Subscriber Identity |
| IP | Internet Protocol |
| ISO | International Organization for Standardization |
| ITU | International Telecommunication Union |

**K**

| | |
|---|---|
| kbps | kilobit per second |

**L**

| | |
|---|---|
| L1 | Layer 1 |
| L2 | Layer 2 |
| L3 | Layer 3 |
| LLME | Lower Layer Management Entity |
| LoCH | Logical Channel |

**M**

| | |
|---|---|
| MAC | Medium Access Control |
| MAI | Multiple Access Interference |

| | |
|---|---|
| Mcps | Mega Chips per Second |
| MLP | MAC Logical Channel Priority |
| MS | Mobile Station |
| MSC | Message Sequence Chart |
| MT | Mobile Terminal |

**N**

| | |
|---|---|
| NACK | Negative Acknowledgement |
| NAS | Non-Access Stratum |
| NL | Network Layer |
| NoACK | No Acknowledgement |
| Nt-SAP | Notification SAP |

**O**

| | |
|---|---|
| OSI | Open Systems Interconnection |
| OVSF | Othogonal Variable Spreading Factor |

**P**

| | |
|---|---|
| PCCH | Paging Control Channel |
| PCH | Paging Channel |
| PCI | Protocol Control Information |
| PCPCH | Physical Common Packet Channel |
| P-CCPCH | Primary Common Control Physical Channel |
| PDCP | Packet Data Convergence Protocol |
| PDSCH | Physical Downlink Shared Channel |
| PDU | Protocol Data Unit |
| PHY | Physical Layer |
| PhCH | Phsical CHannel |
| PID | Process Identifier |
| PICH | Page Indication Channel |
| PL | Physical Layer |
| PRACH | Physical Random Access Channel |
| PUSCH | Physical Uplink Shared Channel |

**Q**

| | |
|---|---|
| QoS | Quality of Service |

**R**

| | |
|---|---|
| RAB | Radio Access Bearer |
| RACH | Random Access Channel |
| RB | Radio Bearer |
| RBM | Radio Bearer Mapping |
| RF | Radio Frame |
| RLC | Radio Link Control |
| RNC | Radio Network Controller |
| RNTI | Radio Network Temporary Identifier |
| RRC | Radio Resource Control |

**S**

| | |
|---|---|
| SAP | Service Access Point |
| S-CCPCH | Secondary Common Control Physical Channel |
| SCH | Synchronization Channel |
| SDL | Specification Description Language |
| SDU | Service Data Unit |
| SF | Spreading Factor |
| SIR | Signal to Interference Ratio |
| SMS | Short Message Service |
| SMTP | Simple Mail Transfer Protocol |
| SPEETCL | SDL Performance Evaluation Tool Class Library |
| SRB | Signalling Radio Bearer |
| SRNC | Serving Radio Network Controller |
| S-RNTI | SRNC-RNTI |

**T**

| | |
|---|---|
| TB | Transport Block |
| TBS | Transport Block Set |
| TCP | Transmission Control Protocol |
| TCTF | Target Channel Type Field |
| TDD | Time Division Duplex |
| TE | Terminal Equipment |
| TF | Transport Format |
| TFC | Transport Format Combination |
| TFCI | Transport Format Combination Indicator |
| TFCS | Transport Format Combination Set |
| TFI | Transport Format Indicator |
| TFS | Transport Format Set |
| TL | Transport Layer |
| TR | Transparent Mode |
| TrCH | Transport Channel |
| TTI | Transmission Time Interval |

**U**

| | |
|---|---|
| UDP | User Datagram Protocol |
| UE | User Equipment |
| UE-Id | User Equipment Identity |
| UM | Unacknowledged Mode |
| UMTS | Universal Mobile Telecommunications System |
| UPDU | URIS PDU |
| URIS | UMTS Radio Interface Simulator |
| URL | Uniform Resource Locator |
| U-RNTI | UTRAN-RNTI |
| USCH | Uplink Shared Channel |
| UTRA | UMTS Terrestrial Radio Access |

UTRAN — UMTS Terrestrial Radio Access Network

**V**

**W**

WCDMA — Wideband Code Division Multiple Access

# BIBLIOGRAPHY

[1] H Holma. A Toskala
*WCDMA for UMTS. Radio Access For Third Generation Mobile Communications*
Wiley. 2000

[2] D Bertsekas, R. Gallager
*Data Networks. 2$^{nd}$ Edition*
Prentice-Hall, 2000

[3] J. Ellsberger, D. Hogrefe. A. Sarma.
*SDL:Formal Object-oriented Language for Communicating Systems*
Prentice-Hall, 1997

[4] B Walke
*Mobile Radio Networks - Networking, Protocols and Traffic Performance. 2$^{nd}$ Edition*
Wiley. 2002

[5] A. Lätsch
*Design and implementation of a simulator for UMTS*
Diplom thesis. Chair of Communication Networks. RWTH-Aachen. 1999
http://www comnets.rwth-aachen.de

[6] M Malkowski
*Simulative Performance Evaluation of UMTS Medium Access Control Protocol for Dedicated Transport Channels*
Diplom thesis. Chair of Communication Networks. RWTH-Aachen. 2002
http://www.comnets.rwth-aachen.de

[7] M. Steppler
*SDL Performance Evaluation Tool Class Library (SPEETCL)*
Chair of Communication Networks, RWTH-Aachen
http //www.comnets.rwth-aachen.de/intern/speetcl/release/doc/html/home html

[8] M. Steppler
*SDL-PR to C++ code generator using the ComNets Class Library*
Chair of Communication Networks, RWTH-Aachen
http //www.comnets.rwth-aachen.de/intern/sdl2cncl/sdl2cncl_toc.html

[9] M Steppler
*Performance Analysis of Communication Systems Formally Specified in SDL*
Chair of Communication Networks, RWTH-Aachen
http://www.steppler.de/beruf/veroeffentlichungen/steppler-1998-wosp.pdf

[10] U. Fornefeld, P. Sievering
*Introduction to SDL*
Chair of Communication Networks, RWTH-Aachen
http //www comnets.rwth-aachen.de/intern/doc/speet/sdl-intro/

[11] *General UMTS Architecture*
3GPP Technical Specification 3G TS 23 101 version 4.0 0, 04/2001
http://www.etsi.org

[12] *UMTS Access Stratum. Services and Functions*
3GPP Technical Specification 3G TS 23.110 version 4.0.0, 04/2001
http://www.etsi.org

[13] *Physical channels and mapping of transport channels onto physical channels*
3GPP Technical Specification 3G TS 25.211 version 4.2.0, 09/2001
http://www etsi.org

[14] *Spreading and modulation (FDD)*
3GPP Technical Specification 3G TS 25.213 version 4.1.0, 06/2001
http://www.etsi.org

[15] *Physical layer procedures (FDD)*
3GPP Technical Specification 3G TS 25.214 version 4.2.0. 09/2001
http //www.etsi.org

[16] *Radio Interface Protocol Architecture*
3GPP Technical Specification 3G TS 25.301 version 4.2 0. 12/2001
http //www.etsi.org

[17] *MAC Protocol Specification*
3GPP Technical Specification 3G TS 25.321 version 4.3.0. 12/2001
http://www.etsi.org

[18] L. Kleinrock, F. Tobagi
*Packet Switching in radio Channels: Part I-Carrier Sense Multiple Access Models and Their Throughtput-Delay Characteristics*
Erricsson Radio Systems, Sweden.
pp.1400-1416, IEEE Transactions on Communication,December-1975.

[19] Z. Dongfeng, L. Bihaiy, Z. Sumin
*Analysis of a Slotted Access Channel with Average Cycle method*
Depratment of Information and Electronics Science, Yunnan University. Kunming, China.

[20] H Olofsson, M. Edvardsson, G. Frank
*Performance Evaluation of Different Random Access Power Ramping Proposals for The WCDMA System*
Erricsson Radio Systems, Sweden.
pp 1505-1509, PIMRC- 99.

[21] Q Zhang, M. Chuah, O. Yue
*Enhanced Power Ramping Scheme for UMTS Random Access Channel*
Bell Laboratories, Lucent Technologies.
pp.2631-2635, IEEE, VTC-99.

[22] J. Moberg, M. Loefgren, R. Karlsson

*Throughput of the WCDMA Random Access Channel*
Radio Communication Systems, Royal Institute of Technology. Stockholm. Sweden.
IST Mobile Communication Summit-2000, Gailway, Ireland.

[23] C Lin
*Investigation of 3rd Generation Mobile Communication RACH Transmission*
Computer and Communication Research Laboratories, Hsinchu, Taiwan
pp.662-667, IEEE, VTC-2000.

[24] P. Choudhary, A Ghosh, L. Jalloul
*Simulated Performance of W-CDMA Random Access Channel*
Advanced Radio Technology Group, Motorola.
pp.2700-2704, IEEE, VTC-2001.

[25] I. Vukovic. T. Brown
*Performance Analysis of the Random Access Channel(RACH) in WCDMA*
Motorola Inc., Arlington Heights, IL-60004.
pp.532-536, IEEE, VTC-2001.